

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Обвінцев О.В., Крєневич А.П., Довгий Б.П., Бородін В.А.,
Гап'як І.В.**

**Задачі з програмування
Мова програмування Python**

Навчальний посібник

КИЇВ – 2021

Наведено задачі з програмування для побудови програм мовою програмування Python. Теми охоплюють структури керування, структури даних, об'єктно-орієнтоване програмування, наукові обчислення, регулярні вирази, використання операційних систем, графічний інтерфейс, програмування у мережі, використання баз даних.

Для студентів, аспірантів і викладачів університетів, а також фахівців з інформаційних технологій, що використовують мову програмування Python.

Навчальне видання

Обвінцев Олександр Вальдемарович

Крєневич Андрій Павлович

Довгий Борис Павлович

Бородін Віктор Анатолійович

Гап'як Ігор Васильович

Зміст

Передмова	5
Тема 1 Лінійні програми.....	6
Тема 2 Розгалужені програми	9
2.1 Умова.....	9
2.2 Бульове присвоєння	10
2.3 Розгалуження.....	12
Тема 3 Циклічні програми.....	14
3.1 Цикли з умовою продовження	14
3.2 Програмування рекурентних співвідношень	16
3.3 Рекурентні обчислення за умовою	22
3.4 Цикли по діапазону значень.....	26
Тема 4 Числові типи даних.....	28
4.1 Арифметика наближених обчислень.....	28
4.2 Цілий тип даних	30
4.3 Дійсний тип даних	32
4.4 Комплексний тип даних.....	36
Тема 5 Символи та рядки.....	36
Тема 6 Списки	41
Тема 7 Кортежі	51
Тема 8 Словники.....	53
Тема 9 Підпрограми.....	59
Тема 10 Модулі	76
Тема 11 Множини.....	81
Тема 12 Файли	83
Тема 13 Класи та об'єкти	91
Тема 14 Рекурсивні структури даних.....	109
Тема 15 Обробка помилок та виключних ситуацій	116

Тема 16 Ітератори та генератори	126
Тема 17 Декоратори	130
Тема 18 Множинне наслідування	134
Тема 19 Метакласи та метапрограмування	140
Тема 20 Наукові обчислення (numpy).....	148
Тема 21 Регулярні вирази (re).....	165
Тема 22 Використання операційних систем (os, io).....	171
Тема 23 Робота з даними у офісних документах	174
Тема 24 Робота з графікою та графічні інтерфейси (tkinter).....	183
Тема 25 Загальна будова глобальних мереж	203
Тема 26 Побудова веб-клієнтів.....	207
Тема 27 Побудова веб-серверів	210
Тема 28 XML та JSON	222
Тема 29 Використання баз даних	236
Тема 30 Тестування. Розповсюдження власних застосунків	248
Тема 31 Паралельні обчислення	257
Проекти для розробки	262
Список джерел	268

Передмова

Цей навчальний посібник призначено для підтримки курсів «Програмування», «Об'єктно-орієнтоване програмування» та «Прикладне програмування», які викладаються на механіко-математичному факультеті Київського національного університету імені Тараса Шевченка. Посібник є продовженням попередніх методичних розробок, виданих співробітниками кафедри математичної фізики, зокрема [1], [2].

Посібник містить більш, ніж 1000 задач з тем курсів. Реальна кількість завдань для студентів є набагато більшою, оскільки багато задач містять підзадачі, позначені літерами а), б) тощо. Частина задач запозичено зі згаданих методичних розробок [1], [2] або з інших джерел ([3], [25]). Значну частину задач розроблено авторами. Задачі орієнтовано на написання програм мовою програмування Python. Але переважна більшість задач може бути використана також для побудови програм іншими сучасними мовами програмування.

Задачі у кожній темі мають різну складність: від найпростіших до більш складних, що дає можливість варіювати рівень складності матеріалу. У кінці посібника наведено перелік програмних проектів для самостійної розробки студентами. Ці проекти є комплексними та містять відомості з декількох тем.

Нумерація тем задач максимально відповідає нумерації тем матеріалів лекцій, які на дату написання цього посібника розташовані на сайті кафедри математичної фізики за адресою <http://matfiz.univ.kiev.ua/>. Але є теми, що охоплюють матеріал інших споріднених курсів для різних навчальних програм (тема 31).

1.7. Обчислити площу трикутника S за трьома сторонами a, b, c .

1.8. Знайти довжини всіх медіан, бісектрис і висот трикутника, якщо відомі три сторони a, b, c .

1.9. Обчислити відстань від точки (x_0, y_0) до:

а) заданої точки (x, y) ;

б) заданої прямої $ax + by + c = 0$;

в) точки перетину прямих $x + by + c = 0$ і $ax + y + c = 0$, де $ab \neq 1$.

1.10. Знайти об'єм циліндра, якщо відомо його радіус основи та висоту.

1.11. Знайти об'єм конуса, якщо відомо його радіус основи та висоту.

1.12. Знайти об'єм тора з внутрішнім радіусом r і зовнішнім радіусом R .

1.13. Ввести ціле число градусів Цельсія C та обчислити й вивести число F – та сама температура в градусах Фаренгейта за формулою $F = 9 * C / 5 + 32$.

1.14. Обчислити силу притягання F між двома тілами, що мають маси m_1, m_2 , на відстані r . Вказівка. Шукана сила визначається за формулою $F = \gamma * m_1 * m_2 / r^2$, де $\gamma = 6.673 * 10^{-11} \text{ Н} * \text{м}^2 / \text{кг}^2$.

1.15. Вважаючи, що Земля має форму сфери радіуса $R = 6350$ км, знайти відстань до лінії горизонту від точки із заданою висотою h над Землею.

1.16. Без попередніх алгебраїчних перетворень обчислити значення арифметичних виразів:

$$\text{а) } y = \frac{c + \frac{a}{a^2 + b^2}}{a + \frac{b}{b^2 + c^2}}; \quad \text{б) } y = 1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4}}}$$

1.17. Наближено визначити значення числа π , використовуючи ланцюговий дріб

$$\pi = 3 + \frac{1}{7 + \frac{1}{15 + \frac{1}{1 + \frac{1}{292}}}}$$

1.18. Наближено визначити період обертання Землі навколо Сонця, використовуючи ланцюговий дріб

$$T = 365 + \frac{1}{4 + \frac{1}{7 + \frac{1}{1 + \frac{1}{3}}}}$$

1.19. Скласти програми для розв'язання рівнянь:

а) $4.2343x + b = c$; б) $(5.23x + b)(d - 2.4y) = 0$;

в) $ax + b = cx + d$, $a - c \neq 0$,

де коефіцієнти a, b, c, d – вводяться з клавіатури.

1.20. Розв'язати квадратне рівняння $ax^2 + bx + c = 0$, де коефіцієнти a, b і c такі, що $b^2 - 4ac > 0$.

1.21. Дано дійсне число x . Користуючись тільки операцією множення, отримати:

- а) x^4 за дві операції; б) x^6 за три операції;
 в) x^9 за чотири операції; г) x^{15} за п'ять операцій;
 ґ) x^{28} за шість операцій; д) x^{64} за шість операцій.

1.22. За найменшу кількість арифметичних операцій, обчислити значення многочлена для введеного з клавіатури значення x :

- а) $y = x^4 + 2x^2 + 1$; б) $y = x^4 + x^3 + x^2 + x + 1$;
 в) $y = x^5 + 5x^4 + 10x^3 + 10x^2 + 5x + 1$; г) $y = x^9 + x^3 + 1$;
 ґ) $y = 16x^4 + 8x^3 + 4x^2 + 2x + 1$; д) $y = x^5 + x^3 + x$.

1.23. Скласти програму для обчислення значення многочлена від двох змінних для введеної з клавіатури пари чисел (x, y) :

- а) $f(x, y) = x^3 + 3x^2y + 3xy^2 + y^3$;
 б) $f(x, y) = x^2y^2 + x^3y^3 + x^4y^4$;
 в) $f(x, y) = x + y + x^2 + y^2 + x^3 + y^3 + x^4 + y^4$.

1.24. Скласти програму взаємного обміну значень цілих змінних x та y :

- а) з використанням додаткової змінної;
- б) без використання додаткової змінної.

1.25. Дано натуральне тризначне число. Знайти:

- а) число одиниць, десятків і сотень цього числа;
- б) суму цифр цього числа;

в) число, утворене при прочитанні заданого числа справа наліво.

1.26. Дано натуральне тризначне число, у якому всі цифри різні. Знайти всі числа, утворені при перестановці цифр заданого числа.

1.27. Від тризначного числа x відняли його останню цифру. Після ділення результату на 10 до частки ліворуч дописали останню цифру числа x та отримали число n . За заданим числом n знайти вихідне число x . Вважати, що $10 < n < 999$, а число десятків у n не дорівнює нулю.

1.28. У тризначному числі x закреслено першу цифру. Якщо отримане число помножити на 10 і добуток додати до першої цифри числа x , то буде отримано число n . За заданим числом n , $1 < n < 999$ знайти число x .

1.29. Тіло починає рухатися без початкової швидкості з прискоренням a . Обчислити:

- а) відстань, яку воно пройде за час t від початку руху;
- б) час, за який тіло досягне швидкості v .

1.30. Обчислити кінетичну енергію тіла масою m , що рухається зі швидкістю v відносно поверхні Землі.

Тема 2 Розгалужені програми

.....

2.1 Умова

2.1. Довести властивості бульових операцій

- а) $p \text{ or } q \equiv q \text{ or } p$;
- б) $(p \text{ and } q) \text{ and } r \equiv p \text{ and } (q \text{ and } r)$;

в) $p \text{ or } q \text{ and } r \equiv (p \text{ or } q) \text{ and } (p \text{ or } r)$;

г) $\text{not } (p \text{ or } q) \equiv \text{not } p \text{ and not } q$.

2.2. Довести властивості умов

а) $x <= y \equiv \text{not } (x > y) \equiv x < y \text{ or } x = y$;

б) $x >= y \equiv \text{not } (x < y) \equiv x > y \text{ or } x = y$.

2.3. Спростити бульові вирази

а) $(x > 0 \text{ or } x = 0)$; б) $(x >= 0 \text{ and } x = 0)$;

в) $\text{not } (x > 0 \text{ and } y > 0)$; г) $\text{not } (x = 0)$;

г) $(x > 0 \text{ or } x >= 0)$; д) $(x > 0 \text{ or } x <= 0)$;

е) $(x > 0 \text{ and } x < 0)$; ж) $x = 0 \text{ or } (x = 0 \text{ and } y > 0)$;

з) $(x > 0 \text{ or } x <= 0) \text{ and } y > 0$;

Розв'язок. д) Користуючись властивостями бульових операцій та відношень, отримаємо $(x > 0 \text{ or } x <= 0) \equiv (x > 0) \text{ or } (x < 0) \text{ or } (x = 0) = (x \neq 0) \text{ or } (x = 0) = \text{True}$.

2.2 Бульове присвоєння

2.4. Записати умову впорядкованості значень змінних a, b та c .

2.5. Перевірити впорядкованість змінних a, b, c :

а) за зростанням їхніх значень;

б) за спаданням їхніх значень;

в) за зростанням чи спаданням їхніх значень.

2.6. Записати умови, що істинні тоді й тільки тоді, коли:

а) натуральне число n – парне;

б) остання цифра числа $n - 0$;

в) ціле число n кратне натуральному числу m ;

г) натуральні числа n і k одночасно кратні натуральному числу m ;

г) сума першої і другої цифри двозначного натурального числа n – двозначне число;

д) число x більше за число y не менше, ніж на 6;

е) принаймні одне з чисел x, y або z більше за 100;

е) тільки одне з чисел x, y або z менше за 1000.

2.7. Скласти програму перевірки можливості існування трикутника із заданими сторонами a, b та c .

Зауваження. У наступних задачах вважати, що межа заданої області самій області не належить.

2.8. Точка площини задана декартовими координатами (x, y) .
Перевірити, чи належить вона координатному квадранту:

а) першому; б) другому; в) третьому; г) четвертому.

2.9. Точка площини задана декартовими координатами (x, y) .
Перевірити, чи належить вона:

а) кільцю, з центром у початку координат, внутрішнім радіусом 1 і зовнішнім 2;

б) квадрату, діагоналі якого перетинаються у початку координат, а одна вершина розташована у точці $(3, 4)$;

в) внутрішності еліпса, фокуси якого розташовані на дійсній осі, велика піввісь еліпса 3, мала піввісь -2 .

2.10. Скласти програму перевірки належності точки (x, y) до зафарбованої області (рис. 2.1.)

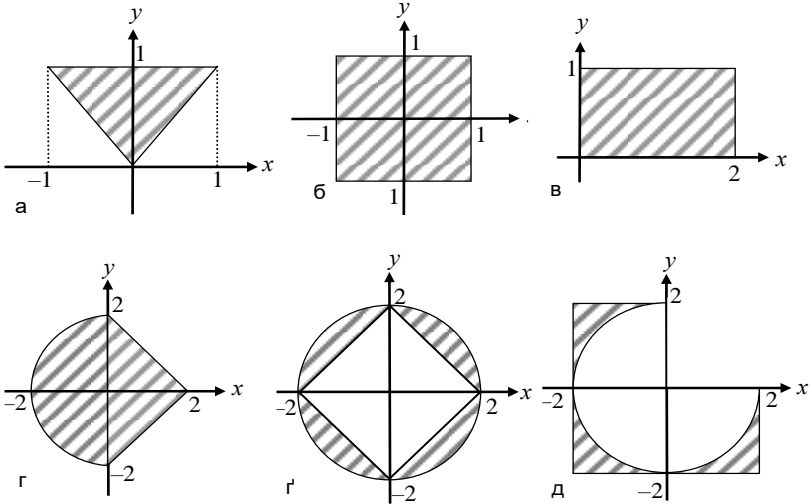


Рис. 2.1

2.11. Створити програму, яка перевіряє, чи належить початок координат трикутнику з вершинами $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$.

2.12. Точка площини задана декартовими координатами (x, y) .
Перевірити, чи належить вона:

а) трикутнику з вершинами $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$;

б) багатокутнику з вершинами $A(x_1, y_1)$, \dots , $A_n(x_n, y_n)$.

2.13. Точка простору задана декартовими координатами (x, y, z) . Перевірити, чи належить вона кулі з радіусом R і центром у початку координат.

2.14. Точка простору задана декартовими координатами (x, y, z) . Перевірити, чи належить вона циліндру, вісь якого збігається з віссю Oz , висота дорівнює h , а нижня основа лежить у площині Oxy та має радіус r .

2.15. Точка простору задана декартовими координатами (x, y, z) . Перевірити, чи належить вона зрізаному конусу, вісь якого збігається з віссю Oz , висота дорівнює h , нижня основа лежить у площині Oxy та має радіус R , а верхня основа – радіус r .

2.16. Точка простору задана декартовими координатами (x, y, z) . Перевірити, чи належить вона внутрішності тора, що утворюється в результаті обертання круга $(x - (a + r))^2 + z^2 \leq r^2$ навколо осі Oz .

2.17. Точка простору задана декартовими координатами (x, y, z) . Перевірити, чи належить вона тетраедру з вершинами у точках $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, $C(x_3, y_3, z_3)$, $D(x_4, y_4, z_4)$.

2.3 Розгалуження

2.18. Визначити більше та менше з двох чисел, введених з клавіатури.

2.19. Дано три дійсних числа. Скласти програму для знаходження числа:

а) найбільшого за модулем; б) найменшого за модулем.

2.20. Дано три дійсних числа x , y і z . Скласти програму для обчислення:

а) $\max(x + y + z, xy - xz + yz, xyz)$; б) $\max(xy, xz, yz)$.

2.21. Дано три дійсних числа x , y і z . Визначити кількість:

а) різних серед них; б) однакових серед них;

в) чисел, що є більшими за їхнє середнє арифметичне значення;

г) чисел, що є більшими за введене з клавіатури число a .

2.22. Обчислити значення функцій:

а) $f(x) = |x|$;

б) $f(x) = \left||x| - 1\right| - 1$;

в) $f(x) = \text{sign}(x)$;

г) $f(x) = \sin|x|$;

$$\text{г) } f(x) = \begin{cases} -x^2 + 1, & x \leq -1, \\ 0, & |x| \leq 1, \\ x^2 - 1, & x > 1; \end{cases} \quad \text{д) } f(x) = \begin{cases} 0, & x \leq 0, \\ x^2, & 0 < x \leq 1, \\ x^4, & x > 1. \end{cases}$$

2.23. Обчислити значення функцій (рис. 2.2.).

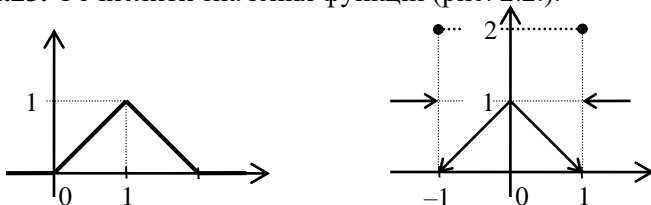


Рис. 2.2

2.24. Обчислити значення виразу:

$$z = \begin{cases} \max(x, y + 5), & x > y, \\ \min(x + 1, y, 3), & x \leq y. \end{cases}$$

2.25. Обчислити значення $x = f(y) - 6.3$, де $y = z + 2$ та

$$f(y) = \begin{cases} y^2 - 0.3, & y < 0, \\ 0, & 0 \leq y \leq 1, \\ y^2 + y, & y > 1. \end{cases}$$

2.26. Перевірити, чи існує трикутник із заданими сторонами a, b, c . Якщо так, то визначити, який він:

а) гострокутний; б) прямокутний; в) тупокутний.

2.27. Скласти програму для розв'язання системи рівнянь при заданому значенні параметра a

$$\begin{cases} y = \max(1, x), \\ y = |ax| - 1. \end{cases}$$

Виконати її при $a = -0.5; a = 1.5; a = 3$.

Вказівка: у програмі необхідно врахувати випадки $a=0$, $0 < |a| < 1$, $1 < |a| \leq 2$, $|a| > 2$.

2.28. Визначити, скільки розв'язків має рівняння та розв'язати його:

а) $ax^2 + bx + c = 0$;

б) $ax^4 + bx^2 + c = 0$.

2.29. Визначити, скільки розв'язків має система рівнянь і розв'язати її:

а)
$$\begin{cases} a_1x + b_1y + c_1 = 0, \\ a_2x + b_2y + c_2 = 0; \end{cases}$$

б)
$$\begin{cases} |x| + |y| = 1, \\ ax + by + c = 0. \end{cases}$$

2.30. Знайти число точок перетину кола $x^2 + y^2 = r^2$ з відрізком $\{x = a, b \leq y \leq b + d^2\}$.

2.31. Скласти програму, яка по колу $(x - u)^2 + (y - v)^2 = r^2$ та прямій $ax + by + c = 0$ встановлює, який випадок має місце:

а) дві точки перетину;

б) одна точка дотику;

в) жодної спільної точки.

2.32. З'ясувати, чи перетинаються два кола на площині.

2.33. Задано два квадрати, сторони яких паралельні координатним осям. З'ясувати, чи перетинаються вони. Якщо так, то знайти координати лівого нижнього та правого верхнього кутів прямокутника, що є їхнім перетином.

2.34. Дано два прямокутники, сторони яких паралельні координатним осям. Відомо координати лівого нижнього та правого верхнього кутів кожного з прямокутників. Знайти координати лівого нижнього та правого верхнього кутів мінімального прямокутника, що містить задані прямокутники.

Тема 3 Циклічні програми

3.1 Цикли з умовою продовження

3.1. Скласти програму обчислення
 $y = \sin(\sin(\dots \sin(x)\dots))$ (n раз).

3.2. Скласти програми для обчислення значень многочленів і виконати їх при заданих значеннях аргументів:

а) $y = x^n + x^{n-1} + \dots + x^2 + x + 1$ $n = 3, x = 2;$

б) $y = x^{2^n} + x^{2^{n-1}} + \dots + x^4 + x^2 + 1$ $n = 4, x = 1;$

в) $y = x^{3^n} + x^{3^{n-1}} + \dots + x^9 + x^3 + 1$ $n = 3, x = 1;$

г) $y = x^{2^n} y^n + x^{2^{n-1}} y^{n-1} + \dots + x^2 y + 1$ $n = 4, x = 1, y = 2;$

д) $y = x^{1^2} + x^{2^2} + \dots + x^{n^2},$ $n = 5, x = -1.$

3.3. Скласти програму обчислення добутку $p = m * n$, використовуючи операцію додавання та виконати її при $m = 5, n = 3$.

3.4. Скласти програму обчислення факторіалу $p = n!$ та подвійного факторіалу $p = n!!$

3.5. Скласти програму обчислення

а) $\sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}$ (n коренів),

б) $\sqrt{3 + \sqrt{6 + \dots + \sqrt{3(n-1)} + \sqrt{3n}}}$.

3.6. Скласти програми обчислення значень многочленів

а) $y = nx^{n-1} + (n-1)x^{n-2} + \dots + 2x + 1, (x < 1, n \geq 0);$

б) $y = \sum_{k=0}^n kx^k (1-x)^{n-k},$ $(0 < x < 1, n \geq 0);$

в) $y = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!},$ (дійсне $x < 1, n \geq 0$).

3.7. Для довільного цілого числа $m \geq 1$ знайти найбільше ціле k , при якому $4^k \leq m$.

3.8. Для заданого натурального числа n одержати найменше число вигляду 2^r , яке перевищує n .

3.9. Знайдіть машинний нуль для вашого компілятора, тобто таке дійсне число $a > 0$, що $1 + a = 1$ буде істиною.

Вказівка: в циклі ділити значення a на 2 доки не виконується вказана вище рівність.

3.10. Ввести послідовність наступним чином: користувачу виводиться напис “a[**]= ”, де замість ** стоїть номер числа, що вводиться. Тобто там виводиться написи “a[0]= ”, і після знаку рівності користувач вводить число, “a[1]= ”, і після знаку рівності користувач вводить число і так далі доки користувач не введе число 0. Після цього потрібно вивести суму введених чисел.

3.11. Визначити із скількох від’ємних чисел починається ненульова послідовність цілих чисел, за якою іде 0.

3.12. Задана непорожня послідовність ненульових цілих чисел, за якою іде 0. Визначити кількість змін знаку в цій послідовності. Наприклад, у послідовності 1, -34, 8,14, -5, 0 знак змінюється три рази.

3.13. Дана непорожня послідовність різних натуральних чисел, за якою слідує 0. Визначити порядковий номер найменшого з них.

3.14. Дана непорожня послідовність різних дійсних чисел, серед яких є хоча б одне від’ємне число, за якою йде 0. Визначити величину найбільшого серед від’ємних членів цієї послідовності.

3.15. Банк пропонує річну ставку по депозиту А та 15% по вкладу додаються до основної суми депозиту кожен рік. Ви кладете в цей банк D гривень. Скільки років потрібно чекати, щоб сума вкладу зросла до очікуваної суми Р?

3.2 Програмування рекурентних співвідношень

3.16. Скласти програми для обчислення елементів послідовностей. Операцію піднесення до степені та функцію обчислення факторіалу не використовувати.

а) $x_k = \frac{x^k}{k} \quad (k \geq 1);$

д) $x_k = \frac{x^{2k}}{(2k)!} \quad (k \geq 0); ;$

б) $x_k = \frac{(-1)^k x^k}{k} \quad (k \geq 1);$

е) $x_k = \frac{x^{2k+1}}{(2k+1)!} \quad (k \geq 0); ;$

в) $x_k = \frac{x^k}{k!} \quad (k \geq 0);$

ж) $x_k = \frac{(-1)^k x^{2k}}{(2k)!} \quad (k \geq 0); ;$

$$\text{г) } x_k = \frac{(-1)^k x^k}{k!} \quad (k \geq 0); \quad \text{з) } x_k = \frac{(-1)^k x^{2k+1}}{(2k+1)!} \quad (k \geq 0);$$

3.17. Числами Фібоначчі називається числова послідовність $\{F_n\}$, задана рекурентним співвідношенням другого порядку $F_0 = 0, F_1 = 1, F_k = F_{k-1} + F_{k-2}, k = 2, 3, \dots$ Скласти програму для обчислення F_n .

3.18. Скласти програму обчислення довільного члена послідовностей, які задані рекурентними співвідношеннями:

$$\begin{aligned} \text{а) } x_n &= x_{n-1} + x_{n-3}, & x_0 &= x_1 = 1, x_2 = 2, & n &= 3, 4, \dots; \\ \text{б) } x_n &= 2x_{n-1} + 3x_{n-3}, & x_0 &= 0, x_1 = 9, & n &= 2, 3, 4, \dots; \\ \text{в) } x_n &= x_{n-1} + x_{n-2} + x_{n-3}, & x_0 &= x_1 = 9, x_2 = 6 & n &= 3, 4, \dots; \\ \text{г) } x_n &= x_{n-1} + 4x_{n-3}, & x_0 &= x_1 = x_2 = 2, & n &= 2, 3, 4, \dots; \\ \text{д) } x_n &= x_{n-1} * (x_{n-2} + 1), & x_0 &= 1, x_1 = 1, & n &= 2, 3, 4, \dots; \\ \text{е) } x_n &= \frac{x_{n-1}}{2} + \frac{3}{4}x_{n-2}, & x_0 &= 0, x_1 = \frac{5}{8}, & n &= 2, 3, 4, \dots \end{aligned}$$

3.19. Скласти програми для обчислення сум:

$$\begin{aligned} \text{а) } S_n &= 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}; \\ \text{б) } S_n &= 1 - \frac{1}{2} + \frac{1}{3} - \dots + (-1)^{n-1} \frac{1}{n}; \\ \text{в) } S_n &= \frac{1}{1*2} + \frac{1}{2*3} + \dots + \frac{1}{(n-1)+n}; \\ \text{г) } S_n &= -\frac{1}{3*1} + \frac{1}{5*3} - \frac{1}{7*5} + \dots + \frac{(-1)^n}{(2n+1)(2n-1)}; \\ \text{д) } S_n &= 1 - 2 - 3 + 4 - 5 - 6 + \dots + (3n-2) - (3n-1) - 3n; \\ \text{е) } S_n &= 1 + 2 + 3 + \dots + n; & \text{ж) } S_n &= a + 2a^2 + 3a^3 + \dots + na^n; \\ \text{з) } S_n &= \sum_{i=1}^n 2^{n-i} i^2; & \text{і) } S_n &= \sum_{i=1}^n C_n^i; & \text{к) } S_n &= \sum_{i=1}^n i C_n^i; \\ \text{л) } S_n &= \sum_{i=1}^n a^i b^{n-i}; & \text{м) } S_n &= \sum_{i=0}^n 2^i i!; & \text{н) } S_n &= \sum_{i=0}^n i!; \end{aligned}$$

$$\text{о) } S_n = \sum_{i=0}^n \frac{1}{i!}; \quad \text{п) } S_n = \frac{1!}{2} + \frac{2!}{2 + \frac{1}{3}} + \dots + \frac{n!}{\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n+1}}.$$

Вказівки. Суму S_n обчислити за допомогою рекурентного співвідношення $S_0 = 0, S_k = S_{k-1} - a_k, k = 1, 2, \dots, n$, де a_k - k -тий доданок.

$$\text{з) } S_i = 2S_{i-1} + i^2; \quad \text{и) } S_n = 2^n; \quad \text{л) } S_n = \frac{a^n - b^n}{a - b}; \quad \text{м) } a_k = 2ka_{k-1}.$$

3.20. Скласти програми для обчислення добутків:

$$\text{а) } P_n = \prod_{i=1}^n \left(2 + \frac{1}{i!}\right); \quad \text{б) } P_n = \prod_{i=1}^n \left(\frac{i+1}{i+2}\right);$$

$$\text{в) } P_n = \prod_{i=1}^n \frac{1}{(i+1)!}; \quad \text{г) } P_n = \prod_{i=1}^n \frac{1}{i^i + 1}.$$

Вказівка. Добуток P_n обчислити за допомогою рекурентного співвідношення $P_0 = 1, P_k = P_{k-1} * a_k, k = 1, 2, \dots, n, k = 1, 2, \dots, n$, де a_k - k - тий множник.

3.21. Скласти програми для обчислення ланцюгових дробів

$$\text{а) } b_n = b + \frac{1}{b + \frac{1}{b + \dots + \frac{1}{b}}}; \quad \text{б) } \lambda_n = 2 + \frac{1}{6 + \frac{1}{10 + \dots + \frac{1}{4n+2}}};$$

$$\text{в) } x_{2n} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{2 + \dots + \frac{1}{2}}}}};$$

Вказівка. Використати рекурентні співвідношення

$$\text{а) } b_0 = b, b_k = b + \frac{1}{b_{k-1}}, k = 1, 2, \dots, n;$$

в) $b_0 = 4n + 2, b_k = 4(n - k) + 2 + \frac{1}{b_{k-1}}, k = 1, 2, \dots, n.$

3.22. Скласти програми для обчислення

а) многочлена Чебишова

$$T_0(x) = 1, T_1(x) = x,$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n = 2, 3, \dots;$$

б) многочлена Ерміта

$$H_0(x) = 1, H_1(x) = 2x,$$

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x), n = 2, 3, \dots$$

даного степеню n в точці x .

3.23. Скласти програми обчислення довільного елемента послідовностей, заданих рекурентними співвідношеннями

а) $v_0 = 1, v_1 = 0.3, v_i = (i + 2)v_{i-2}, i = 2, 3, \dots$

б) $v_0 = v_1 = v_2 = 1, v_i = (i + 4)(v_{i-1} - 1) + (i + 5)v_{i-3}, i = 3, 4, \dots$

в) $v_0 = v_1 = 0, v_2 = \frac{3}{2}, v_i = \frac{i-2}{(i-3)^2 + 1}v_{i-1} - v_{i-2}v_{i-3} + 1, i = 2, 3, \dots$

3.24. Скласти програму обчислення довільного елемента послідовності v_n , визначеної системою

співвідношень

$$v_0 = v_1 = 1, v_i = \frac{u_{i-1} - v_{i-1}}{|u_{i-2} + v_{i-1}| + 2}, i = 2, 3, \dots;$$

$$\text{де } u_0 = u_1 = 0, u_i = \frac{u_{i-1} - u_{i-2}v_{i-1} - v_{i-2}}{1 + u_{i-1}^2 + v_{i-1}^2}, i = 2, 3, \dots;$$

3.25. Скласти програми для обчислення сумм

а) $S_n = \sum_{k=1}^n 2^k a_k$, де $a_1 = 0, a_2 = 1, a_k = a_{k-1} + k * a_{k-2}, k = 3, 4, \dots;$

б) $S_n = \sum_{k=1}^n \frac{3^k}{a_k}$, де $a_1 = a_2 = 1, a_k = \frac{a_{k-1}}{k} + a_{k-2}, k = 3, 4, \dots;$

в) $S_n = \sum_{k=1}^n \frac{k!}{a_k}$, де $a_1 = a_2 = 1, a_k = a_{k-1} + \frac{a_{k-1}}{2^k}, k = 3, 4, \dots;$

$$\text{г) } S_n = \sum_{k=1}^n k! a_k, \text{ де } a_1 = 0, a_2 = 1, a_k = a_{k-1} + \frac{a_{k-2}}{(k-1)!}, k = 3, 4, \dots;$$

$$\text{р) } S_n = \sum_{k=1}^n \frac{a_k}{2^k}, \text{ де } a_1 = a_2 = a_3 = 1, a_k = a_{k-1} + a_{k-3}, k = 4, 5, \dots;$$

$$\text{д) } S_n = \sum_{k=1}^n \frac{2^k}{k!} a_k, \text{ де } a_0 = 1, a_k = k a_{k-1} + \frac{1}{k}, k = 1, 2, \dots$$

Вказівка. Позначимо загальний член ряду через b_k . Послідовність a_k задається залежностями вигляду (R_1) для е), (R_2) для а)–г) та (R_3) для д); $S_k = g(S_{k-1}, b_k)$. Значення a_k будуть обчислюватись за теоремами 1–2. Для обчислення послідовності S_k цикли доповнюються однією змінною.

3.26. Скласти програми для обчислення сумм

$$\text{а) } S_n = \sum_{k=1}^n \frac{2^k}{a_k + b_k},$$

$$\text{де } \begin{cases} a_1 = 0, a_2 = 1, \\ a_k = \frac{a_{k-1}}{k} + a_{k-2} b_k, \end{cases} \quad \begin{cases} b_1 = 0, b_2 = 1, \\ b_k = b_{k-1} + a_{k-1}, \end{cases} \quad k = 3, 4, \dots;$$

$$\text{б) } S_n = \sum_{k=1}^n \frac{a_k b_k}{(k+1)!},$$

$$\text{де } \begin{cases} a_1 = u, \\ a_k = 2b_{k-1} + a_{k-1}, \end{cases} \quad \begin{cases} b_1 = v, \\ b_k = 2a_{k-1}^2 + b_{k-1}, \end{cases} \quad k = 2, 3, \dots;$$

u, v - задані дійсні числа;

$$\text{в) } S_n = \sum_{k=1}^n \frac{2^k}{(1 + a_k^2 + b_k^2) k!},$$

$$\text{де } \begin{cases} a_1 = 1, \\ a_k = 3b_{k-1} + 2a_{k-1}, \end{cases} \quad \begin{cases} b_1 = 1, \\ b_k = 2a_{k-1} + b_{k-1}, \end{cases} \quad k = 2, 3, \dots;$$

$$\text{г) } S_n = \sum_{k=1}^n \left(\frac{a_k}{b_k} \right)^k,$$

$$\text{де } \begin{cases} a_0 = 1, a_1 = 2, \\ a_k = b_{k-2} + \frac{b_k}{2}, \end{cases} \quad \begin{cases} a_0 = 5, b_1 = 5, \\ b_k = b_{k-2}^2 - a_{k-1}, \end{cases} \quad k = 2, 3, \dots;$$

$$\text{д) } S_n = \sum_{k=1}^n \frac{a_k}{1 + b_k},$$

$$\text{де } \begin{cases} a_0 = 1, \\ a_k = b_{k-1} a_{k-1}, \end{cases} \quad \begin{cases} b_0 = 1, \\ b_k = b_{k-1} + a_{k-1}, \end{cases} \quad k = 1, 2, \dots$$

3.27. Скласти програми для обчислення добутоків

$$\text{а) } P_n = \prod_{k=0}^n \frac{a_k}{3^k}, \quad \text{де } \begin{cases} a_0 = a_1 = 1, a_2 = 3, \\ a_k = a_{k-3} + \frac{a_{k-2}}{2^{k-1}}, \end{cases} \quad k = 3, 4, \dots;$$

$$\text{б) } P_n = \prod_{k=1}^n a_k b_k,$$

$$\text{де } \begin{cases} a_1 = 1, \\ a_k = (\sqrt{b_{k-1}} + a_{k-1}) / 5, \end{cases} \quad \begin{cases} b_1 = 1, \\ b_k = 2b_{k-1} + 5a_{k-1}^2, \end{cases}$$

$$k = 2, 3, \dots$$

3.28. Скласти програми для обчислення визначників порядку

n :

$$\text{а) } \begin{vmatrix} 2 & 3 & 0 & \dots & 0 \\ 1 & 2 & 3 & \dots & 0 \\ 0 & 1 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2 \end{vmatrix};$$

$$\text{б) } \begin{vmatrix} 2 & 1 & 0 & \dots & 0 \\ 1 & 2 & 1 & \dots & 0 \\ 0 & 1 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2 \end{vmatrix};$$

$$\text{в) } \begin{vmatrix} 3 & 2 & 0 & \dots & 0 \\ 1 & 3 & 2 & \dots & 0 \\ 0 & 1 & 3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 3 \end{vmatrix};$$

$$\text{г) } \begin{vmatrix} 7 & 5 & 0 & \dots & 0 \\ 2 & 7 & 5 & \dots & 0 \\ 0 & 2 & 7 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 7 \end{vmatrix};$$

$$\text{д) } \begin{vmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 \end{vmatrix};$$

$$е) \begin{vmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{vmatrix};$$

$$є) \begin{vmatrix} a+b & ab & 0 & 0 & \dots & 0 \\ 1 & a+b & ab & 0 & \dots & 0 \\ 0 & 1 & a+b & ab & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & a+b \end{vmatrix};$$

$$ж) \begin{vmatrix} 1 & 2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 3 & 4 & 3 & 0 & 0 & \dots & 0 & 0 \\ 0 & 2 & 5 & 3 & 0 & \dots & 0 & 0 \\ 0 & 0 & 2 & 5 & 3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 5 & 3 \\ 0 & 0 & 0 & 0 & 0 & \dots & 2 & 5 \end{vmatrix};$$

$$з) \begin{vmatrix} a & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & a & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & a & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & a \end{vmatrix};$$

$$і) \begin{vmatrix} 1+x^2 & x & 0 & 0 & \dots & 0 & 0 \\ x & 1+x^2 & x & 0 & \dots & 0 & 0 \\ 0 & x & 1+x^2 & x & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & x & 1+x^2 \end{vmatrix}.$$

3.3 Рекурентні обчислення за умовою

3.29. Маємо дійсне число a . Скласти програми обчислення:

а) серед чисел $1, 1 + \frac{1}{2}, 1 + \frac{1}{2} + \frac{1}{3}, \dots$ першого, більшого за a ;

б) такого найменшого n , що $1 + \frac{1}{2} + \dots + \frac{1}{n} > a$.

3.30. Скласти програми обчислення:

а) номера найбільшого числа Фібоначчі, яке не перевищує задане число a ;

б) номера найменшого числа Фібоначчі, яке більше заданого числа a ;

в) суми всіх чисел Фібоначчі, які не перевищують 1000.

3.31. Дана непорожня послідовність з натуральних чисел, за якою йде 0. Обчислити суму тих з них, порядкові номери яких - числа Фібоначчі.

3.32. Скласти програми для обчислення найменшого додатнього члена числових послідовностей, які задаються рекурентними співвідношеннями, та його номера

а) $x_n = x_{n-1} + x_{n-2} + 100, \quad x_1 = x_2 = -99, \quad n = 3, 4, \dots;$

б) $x_n = x_{n-1} + x_{n-2} + x_{n-3} + 200, \quad x_1 = x_2 = x_3 = -99, \quad n = 4, 5, \dots;$

в) $x_n = x_{n-1} + x_{n-3} + 100, \quad x_1 = x_2 = x_3 = -99, \quad n = 4, 5, \dots$

3.33. Скласти програму, яка з'ясовує, чи входить задана цифра до запису заданого натурального числа.

3.34. Скласти програму "обернення" (запису в оберненому порядку цифр) заданого натурального числа.

Вказівка. Для побудови числа використати рекурентне співвідношення $y_0 = 0, y_i = y_{i-1} * 10 + a_i$, де a_i - наступна цифра числа n при розгляді цифр справа наліво.

3.35. Скласти програму, яка визначає потрібний спосіб розміну будь-якої суми грошей до 99 коп. за допомогою монет вартістю 1, 2, 5, 10, 25, 50 коп.

3.36. Скласти програми наближеного обчислення суми всіх доданків, абсолютна величина яких не менше $\varepsilon > 0$:

а) $y = \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots;$

б) $y = \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots;$

в) $y = \operatorname{sh} x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots;$

г) $y = \operatorname{ch} x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots;$

д) $y = e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots;$

е) $y = \ln(1+x) = x - \frac{x^2}{2!} + \frac{x^3}{3!} - \dots, \quad (|x| < 1);$

$$\text{ж) } y = \frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots, \quad (|x| < 1);$$

$$\text{з) } y = \ln \frac{1+x}{1-x} = 2 * \left[\frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right], \quad (|x| < 1);$$

$$\text{и) } y = \frac{1}{(1+x)^2} = 1 - 2 * x + 3 * x^2 - \dots, \quad (|x| < 1);$$

$$\text{к) } y = \frac{1}{(1+x)^3} = 1 - \frac{2 * 3}{2} x + \frac{3 * 4}{2} x^2 - \frac{4 * 5}{2} x^3 + \dots, \quad (|x| < 1);$$

$$\text{л) } y = \frac{1}{1+x^2} = 1 - x^2 + x^4 - x^6 + \dots, \quad (|x| < 1);$$

$$\text{м) } y = \sqrt{1+x} = 1 + \frac{1}{2} x - \frac{1}{2 * 4} x^2 + \frac{1 * 3}{2 * 4 * 6} x^3 - \dots, \quad (|x| < 1);$$

$$\text{н) } y = \frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} x + \frac{1 * 3}{2 * 4} x^2 - \frac{1 * 3 * 5}{2 * 4 * 6} x^3 - \dots, \quad (|x| < 1);$$

$$\text{о) } y = \arcsin x = x + \frac{1}{2} \frac{x^3}{3!} + \frac{1 * 3}{2 * 4} \frac{x^5}{5!} + \dots, \quad (|x| < 1).$$

Вказівка. Суму y обчислювати за допомогою рекурентного співвідношення $S_0 = 0, S_k = S_{k-1} + a_k, k = 1, 2, \dots$, де $a_k - k$ -тий доданок, для обчислення якого також складається рекурентне співвідношення. В якості умови повторення циклу розглядається умова $|a_k| \geq \varepsilon$.

3.37. Маємо дійсні числа x, ε ($x \neq 0, \varepsilon > 0$). Обчислити з точністю ε нескінченну суму і вказати кількість врахованих доданків.

$$\begin{array}{ll} \text{а) } \sum_{k=0}^{\infty} \frac{x^{2k}}{2k!}; & \text{б) } \sum_{k=0}^{\infty} \frac{(-1)^k x^k}{(k+1)^2}; \\ \text{в) } \sum_{k=0}^{\infty} \frac{x^{2k}}{2^k k!}; & \text{г) } \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{k!(2k+1)!}. \end{array}$$

3.38. Маємо ціле $n * 2$. Скласти програму для обчислення всіх простих чисел з діапазону $[2, n]$.

3.39. Скласти програму друку всіх простих дільників заданого натурального числа.

3.40. Скласти програму, яка визначає чи є задане натуральне число n досконалим, тобто рівним сумі всіх своїх (додатніх) дільників, крім самого цього числа (наприклад, число 6 - досконале: $6=1+2+3$).

Вказівка. Шукаємо суму S всіх дільників заданого числа n . Якщо $S=n$, то число, яке перевіряємо, є досконалим. Перша ідея полягає в знаходженні дільників числа n в діапазоні $[1, n \text{ div } 2]$. У відповідності з другою ідеєю пошук ведеться тільки між 1 та \sqrt{n} і якщо дільник знайдений, то до суми S додаються як дільник, так і частка.

3.41. Дано натуральне число k . Скласти програму одержання k -тої цифри послідовності

а) 110100100010000 ..., в якій виписані підряд степені 10;

б) 123456789101112 ..., в якій виписані підряд всі натуральні числа;

в) 149162536 ..., в якій виписані підряд квадрати всіх натуральних чисел;

г) 01123581321 ..., в якій виписані підряд всі числа Фібоначчі.

3.42. Скласти програму знаходження кореня рівняння $tgx = x$ на відрізку $[0,001;1,5]$ із заданою точністю ε , використовуючи метод ділення відрізка навпіл.

3.43. Знайти корінь рівняння $x^3 + 4x^2 + x - 6 = 0$, який міститься на відрізку $[0,2]$, з заданою точністю

Вказівка. Одним з методів розв'язування рівняння є метод хорд, який полягає в обчисленні елементів послідовності

$$u_0 = a,$$

$$u_n = u_{n-1} - \frac{y(u_{n-1})}{y(b) - y(u_{n-1})} (b - u_{n-1})$$

до виконання умови $|u_n - u_{n-1}| < \varepsilon$. В умовах нашої задачі

$$a = 0, b = 2, y(x) = x^3 + 4x^2 + x - 6.$$

3.4 Цикли по діапазону значень

3.44. Вивести на екран такий рядок:

$$n! = 1*2*3*4*5*...*n$$

де n – введене з клавіатури число.

3.45. Вивести на екран таблицю множення на 5:

$$1 \times 5 = 5$$

$$2 \times 5 = 10$$

...

$$9 \times 5 = 45$$

3.46. Вивести на екран таблицю:

$$1 \quad | \quad 2 \quad | \quad 3 \quad | \quad \dots \quad | \quad n-1 \quad | \quad n$$

----- ... -----

$$a+1 \quad | \quad a+2 \quad | \quad a+3 \quad | \quad \dots \quad | \quad a+n-1 \quad | \quad a+n$$

3.47. Написати програму обчислення добутку двох натуральних чисел, використовуючи лише операцію додавання.

3.48. Дано натуральне число n . Написати програми обчислення значень виразів при заданому значенні x :

а) $1 + (x-1) + (x-1)^2 + \dots + (x-1)^n$;

б) $1 + \frac{1}{x^2+1} + \frac{1}{(x^2+1)^2} + \dots + \frac{1}{(x^2+1)^n}$;

в) $x + (2x)^2 + \dots + ((n-1)x)^{n-1} + (nx)^n$;

г) $1 + \sin x + \sin^2 x + \dots + \sin^n x$.

3.49. Задані натуральне число n , дійсні числа a_1, a_2, \dots, a_n .

Скласти програму для знаходження:

а) $\max(a_1, a_2, \dots, a_n)$; б) $\min(a_1, a_2, \dots, a_n)$;

в) $\max(a_2, a_4, \dots)$; г) $\min(a_1, a_3, \dots)$;

д) $\min(a_2, a_4, \dots) + \max(a_1, a_3, \dots)$;

е) $\max(|a_1|, \dots, |a_n|)$; ж) $\max(-a_1, a_2, -a_3, \dots, (-1)^n a_n)$;

$$з) (\min(a_1, \dots, a_n))^2 - \min(a_{1n}^2, \dots, a_n^2).$$

3.50. Дано натуральне число n , цілі числа a_1, a_2, \dots, a_n .

Скласти програму знаходження

а) $\min(a_1, 2a_2, \dots, na_n)$;

б) $\min(a_1 + a_2, \dots, a_{n-1} + a_n)$;

в) $\max(a_1, a_1 a_2, \dots, a_1 a_2 \dots a_n)$;

г) кількості парних серед a_1, a_2, \dots, a_n ;

д) кількості повних квадратів серед a_1, a_2, \dots, a_n ;

е) кількості квадратів непарних чисел серед a_1, a_2, \dots, a_n .

3.51. Дано натуральне число n . Скласти програму обчислення факторіала $y=n!$, використовуючи

а) цикл по діапазону із зростанням;

б) цикл по діапазону зі спаданням.

3.52. Скласти програму обчислення подвійного факторіала натурального числа n $y = n!!$.

Вказівка. За означенням

$$n!! = \begin{cases} 1 \cdot 3 \cdot 5 \cdot \dots \cdot n, & \text{якщо } n - \text{непарне,} \\ 2 \cdot 4 \cdot 6 \cdot \dots \cdot n, & \text{якщо } n - \text{парне.} \end{cases}$$

3.53. Скласти програми обчислення факторіалів:

а) $y = (2n)!!$; б) $y = (2n+1)!!$; в) $y = n!n!!(n+1)!!$.

3.54. Задане натуральне число n . Скласти програми обчислення добутків

а) $p = \left(1 + \frac{1}{1^2}\right) \left(1 + \frac{1}{2^2}\right) \dots \left(1 + \frac{1}{n^2}\right), \quad n > 2$;

б) $p = \left(1 - \frac{1}{2^2}\right) \left(1 - \frac{1}{3^2}\right) \dots \left(1 + \frac{1}{n^2}\right), \quad n > 2$.

3.55. Скласти програму друку таблиці значень функції $y = \sin x$ на відрізку $[0, 1]$ з кроком $h = 0.1$.

3.56. Скласти програму визначення кількості тризначних натуральних чисел, сума цифр яких дорівнює n ($n > 1$). Операцію ділення не використовувати.

3.57. Дано n цілих чисел. Скласти програму, що визначає, скільки з них більші за своїх "сусідів", тобто попереднього та наступного чисел.

3.58. Задані натуральне число n , дійсні числа y_1, \dots, y_n . Скласти програму визначення

$$\text{а) } \max(|z_1|, \dots, |z_n|), \quad \text{де } z_i = \begin{cases} y_i, & \text{при } |y_i| \leq 2, \\ 0.5, & \text{у інших випадках} \end{cases} ;$$

$$\text{б) } \min(|z_1|, \dots, |z_n|), \quad \text{де } z_i = \begin{cases} y_i, & \text{при } |y_i| \geq 1, \\ 2, & \text{у інших випадках} \end{cases} ;$$

$$\text{в) } z_1 + z_2 + \dots + z_n, \quad \text{де } z_i = \begin{cases} y_i, & \text{при } 0 < y_i < 10, \\ 1, & \text{у інших випадках} \end{cases} .$$

Тема 4 Числові типи даних

4.1 Арифметика наближених обчислень

4.1. Обчислити значення функції *round*

а) $\text{round}(35.774873, 5, 10)$;

б) $\text{round}(0.0000237927, 4, 10)$;

в) $\text{round}((110.0100110)_2, 7, 2)$;

г) $\text{round}((3.77673)_8, 3, 8)$;

д) $\text{round}(1345271, 4, 10)$; $\text{round}(1345271, 4, 10)$;

е) $\text{round}((21.4322)_5, 3, 5)$.

4.2. Підібрати приклади, що показують порушення співвідношень

а) $x +_p (y +_p z) = (x +_p y) +_p z$;

б) $x *_p (y *_p z) = (x *_p y) *_p z$;

в) $x *_p (y +_p z) = x *_p y +_p x *_p z$; $x *_p (y +_p z) = x *_p y +_p x *_p z$

при $p = 8$.

Відповідь

a) $x=11111113$; $y=-111111$., $z=-7.5111111$. $x=1111111$

4.3. Вкажіть x такий, що $(x+_p x)/_p 2 <> x$ при $p=8$.

4.4. Чи виконується рівність $x/_p y = x*_p (1/_p y)$ для всіх дійсних чисел?

4.5. Доведіть або спростуйте співвідношення

a) $0-_p (0-_p x) = x$;

b) $1/_p (1/_p x) = x$.

4.6. Чи вірно, що $u \leq (u+_p v)_p / 2 \leq v$ при $u \leq v$?

Розв'язок. В звичайній арифметиці при $u \leq v$ справедлива нерівність

$$u = \frac{1}{2}(u+u) \leq \frac{1}{2}(u+v) \leq \frac{1}{2}(v+v) = v.$$

В арифметиці наближених обчислень $x=(x+_p x)/_p 2$ справедливе не завжди, що призводить до порушення нерівностей $u \leq (u+_p v)_p / 2 \leq v$ при $u \leq v$. Достатньо взяти ($p=8$) $u=9.2222222$, $v=9.2222223$.

4.7. Нехай $p=4$. Послідовність a_n ($n \geq 1$) задана співвідношенням $a_1=1000$, $a_n=0.5$ ($n \geq 2$). Скласти програму обчислення суми $a_1+a_2+\dots+a_n$ з точністю $\varepsilon=1$.

4.8. Нехай $p=4$. Послідовність a_i ($i \geq 1$) задана співвідношенням $a_i=5000+i$. Скласти програму обчислення середнього арифметичного $\frac{1}{n}(a_1+\dots+a_n)$ з точністю $\varepsilon=1$.

4.9. Розв'язати квадратне рівняння $ax^2+bx+c=0$, використовуючи наближені обчислення з чотирма ($p=4$) і шістьма ($p=6$) значущими цифрами. Порівняти отримані корені один з одним і з точним розв'язком. Знайти спосіб обчислення і скласти програму, що дає результат з відносною похибкою δ .

Вказівка: Скористатися теоремою Вієта.

4.2 Цілий тип даних

4.10. Скласти програму обчислення найбільшого спільного дільника

а) двох натуральних чисел;

б) двох цілих чисел

за допомогою метода Евкліда.

4.11. Скласти програму обчислення Метод для урівнення найменшого спільного кратного двох натуральних чисел m і n . **4.12.** Дано натуральні числа m і n . Знайти такі натуральні числа p і q , не виключаючи спільних дільників, що $p/q = m/n$.

4.13. Знайти всі прості нескоротні дроби, що містяться між 0 і 1, знаменники яких не перевищують 7 (дріб задається двома числами – чисельником та знаменником).

4.14. Дано натуральне число n . Скласти програму знаходження всіх таких натуральних q , що n ділиться на q^2 і не ділиться на q^3 .

4.15. Дано натуральні числа m і n . Скласти програму знаходження всіх їх натуральних спільних кратних, менших добутка $m*n$.

4.16. Дано натуральні числа m і n . Скласти програму знаходження всіх їх спільних дільників.

4.17. Два натуральних числа називаються дружніми, якщо кожне з них дорівнює сумі всіх дільників іншого, крім самого цього числа. Скласти програму знаходження всіх пар дружніх чисел, що лежать в діапазоні від 200 до 300.

4.18. Дано натуральне число n . Скласти програму знаходження всіх піфагорових трійок натуральних чисел, кожне з яких не перевищує n , тобто всіх таких трійок натуральних чисел a, b, c , що $a^2 + b^2 = c^2$ ($a \leq b \leq c \leq n$).

4.19. Натуральне число з n цифр є числом Армстронга, якщо сума його цифр, піднесених до n -того степеня, дорівнює самому числу (наприклад, $153 = 1^3 + 5^3 + 3^3$). Скласти програму знаходження всіх чисел Армстронга, що складаються з двох, трьох та чотирьох цифр.

4.20. Дано натуральне число n . Скласти програму, що визначає, чи можна подати його у вигляді суми двох квадратів натуральних чисел. Якщо це можливо, то

- а) вказати пару a, b таких натуральних чисел, що $n = a^2 + b^2$;
- б) вказати пару a, b таких натуральних чисел, що $n = a^2 + b^2$, $a \geq b$.

4.21. Дано натуральне число n . Скласти програму, що дозволяє

- а) переставити першу та останню цифри числа n ;
- б) приписати по одиниці в початок та кінець запису числа n ;
- в) одержати суму m останніх цифр числа n .

4.22. Дано натуральне число n . Скласти програму, що визначає серед чисел $1, \dots, n$ всі такі, запис яких співпадає з останніми цифрами запису їх квадратів (напр., $6^2 = 36$, $25^2 = 625$ тощо).

4.23. Назвемо натуральне число паліндромом, якщо його запис читається однаково зліва направо і справа наліво (наприклад, $1, 393, 4884$). Скласти програму, що визначає, чи є задане натуральне число n паліндромом.

4.24. Дано натуральне число n . Скласти програму знаходження всіх менших n натуральних чисел

- а) які при піднесенні до квадрату дають паліндром;
- б) паліндромів, які при піднесенні до квадрату дають також паліндром.

4.25. Задані натуральне число n , цілі числа a, x_1, \dots, x_n . Скласти програму визначення номера члена послідовності, який дорівнює a . Якщо такого немає, то відповіддю має бути число 0 .

4.26. Задані натуральне число n , цілі числа a, x_1, \dots, x_n . Якщо в послідовності x_1, \dots, x_n є хоча б один член, що дорівнює a , то отримати суму всіх членів, які слідуєть за таким членом послідовності, в протилежному випадку відповіддю має бути відповідне текстове повідомлення.

4.27. Задані натуральне число n , цілі числа a_1, \dots, a_n . Отримати суму додатніх, кількість від'ємних і число нульових членів послідовності a_1, \dots, a_n .

4.28. Задані натуральні числа n і p , цілі числа a_1, \dots, a_n . Скласти програму обчислення добутку членів послідовності, кратних p .

4.29. Дано натуральне число n . Скласти програму, яка з'ясовує, чи можна подати $n!$ у вигляді добутку трьох послідовних чисел.

4.3 Дійсний тип даних

4.30. Задані рекурентні співвідношення

$$a_0 = 1, c_0 = 1 - x,$$

$$a_i = a_{i-1} * (c_i + 1), c_i = c_{i-1}^2, i = 1, 2, \dots$$

Довести, що $\lim_{n \rightarrow \infty} a_n = \frac{1}{x}, (0 < x < 2)$ та скласти програму

обчислення величини $\frac{1}{x}$, що не використовує операцію ділення

Вказівка. Скористатись співвідношеннями $c_n = c_0^{2^n}$ і

$a_n = \frac{1 - c_n}{x}$, останнє з яких одержується з рівностей

$$a_n = (1 + c_{n-1}) * \dots * (1 + c_1) * (1 + c_0), \quad \frac{1 + c_{i-1}}{1 - c_i} = \frac{1}{1 - c_{i-1}}.$$

4.31. Скласти програму обчислення із заданою точністю границь послідовностей, утворених за законами:

а) $a_n = \frac{n}{\sqrt{n^2 + 1} + \sqrt{n^2 - 1}}$; б) $a_n = \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{3}\right) \cdot \dots \cdot \left(1 - \frac{1}{n+1}\right)$;

в) $a_n = \left(1 - \frac{1}{2!}\right) \cdot \left(1 + \frac{1}{3!}\right) \cdot \dots \cdot \left(1 + \frac{(-1)^n}{(n+1)!}\right)$.

4.32. Для заданих $\varepsilon > 0$ і $x > 0$ скласти програму наближеного обчислення кореня k -го степеня $y(x) = \sqrt[k]{x}$ з числа x , використовуючи відповідні рекурентні співвідношення і границю $\lim_{n \rightarrow \infty} x_n = \sqrt[k]{x}$.

а) корінь квадратний $y(x) = \sqrt{x}, k = 2$:

$$c = 1 + \frac{1}{1 + \frac{1}{1 + \dots}}$$

4.34. Скласти програму обчислення інтегралу

$$\int_0^x e^{-x^2} dx = x - \frac{x^3}{3 \cdot 1!} + \frac{x^5}{5 \cdot 2!} - \frac{x^7}{7 \cdot 3!} + \dots$$

з заданою точністю.

4.35. Скласти програму наближеного обчислення із заданою точністю границі послідовності, заданої рекурентним співвідношенням

а) $a_0 = 0, a_i = \frac{a_{i-1} + 1}{a_{i-1} + 2}, i \geq 1;$ б) $a_0 = 1, a_i = \frac{2 - a_{i-1}^3}{5}, i \geq 1.$

4.36. Задано дійсне число x . Послідовність $a_1 a_2, \dots$ утворена за законом $a_1 = x$: далі для $i \geq 2$ виконано:

а) $a_i = \sqrt{4 \cdot a_{i-1}^2 - 2x};$ б) $a_i = \frac{16 + x}{1 + |a_{i-1}^3|};$
 в) $a_i = 2 \cdot a_{i-1} + \frac{x}{4 + a_{i-1}^2};$ г) $a_i = 3 + \frac{1}{2^i} \cos^2(a_{i-1} - x).$

Скласти програму для обчислення границі послідовності a_n із заданою точністю.

4.37. Задано рекурентні співвідношення

$$x_1 = a, y_1 = b,$$

$$x_i = \frac{1}{2}(x_{i-1} + y_{i-1}), y_i = \sqrt{x_{i-1} \cdot y_{i-1}}, i \geq 2 (a > b > 0).$$

Скласти програму для обчислення першого члена послідовності x_i такого, що $|x_i - y_i| < \varepsilon, (\varepsilon > 0).$

4.38. Скласти програму наближеного обчислення числа π :

а) за формулою Грегорі

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots;$$

б) використовуючи добуток

$$\frac{2}{\pi} = \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}} \cdot \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}} \cdot \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}} \cdot \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}} \cdot \sqrt{\frac{1}{2}} \cdot \dots;$$

г) за формулою Валліса

$$\frac{\pi}{2} = \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdot 8 \cdot 8}{1 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 7 \cdot 7 \cdot 9} \dots$$

Проаналізувати, який з методів дає кращий результат за меншу кількість ітерацій.

4.39. Дано n дійсних чисел. Скласти програму для знаходження того з них, що є найближчим до цілого числа.

4.40. Задано дійсні числа x, y ($x > 0, y > 1$). Скласти програму обчислення цілого числа k (додатного, від'ємного або рівного нулю), що задовольняє умову $y^{k-1} \leq x < y^k$.

4.41. Задано послідовність дійсних чисел x_1, x_2, \dots, x_n (n заздалегідь невідоме), за якою йде 0. Скласти програму для обчислення суми

$$y = nx_1 + (n-1)x_2 + \dots + 2x_{n-1} + x_n.$$

4.42. Задано послідовність із n дійсних чисел x_1, x_2, \dots, x_n ($n \geq 3$). Скласти програму для обчислення:

а) $(x_1 + 2x_2 + x_3) \cdot (x_2 + 2x_3 + x_4) \cdot \dots \cdot (x_{n-2} + 2x_{n-1} + x_n)$;

б) $(x_1 + x_2 + x_3) \cdot x_2 + \dots + (x_{n-2} + x_{n-1} + x_n) \cdot x_{n-1}$.

4.43. Задано натуральне число n , дійсні числа a_1, a_2, \dots, a_n . Скласти програму, що визначає в послідовності a_1, a_2, \dots, a_n кількість сусідств:

а) двох додатних чисел; б) двох чисел різних знаків.

4.44. Скласти програму обчислення суми

$$1 - \frac{1}{2} + \frac{1}{3} - \dots + \frac{1}{9999} - \frac{1}{10000}$$

чотирма способами:

- а) послідовно зліва направо;
- б) послідовно зліва направо обчислюються окремо суми додатних і від'ємних доданків, далі ці суми складаються;
- в) послідовно справа наліво;
- г) послідовно справа наліво обчислюються окремо суми додатних і від'ємних доданків, далі ці суми складаються.

Порівняти результат із даним значенням суми (30 десяткових знаків):

$$0.693097183059945296917232371458\dots$$

Який спосіб дає найменшу похибку обчислень, чому?

4.4 Комплексний тип даних

4.45. Визначити програми для обчислення

- а) аргументу;
- б) модуля комплексного числа.

4.46. Скласти програму обчислення коренів квадратного рівняння з заданими комплексними коефіцієнтами.

4.47. Скласти програму обчислення значень квадратного тричлена з комплексними коефіцієнтами в заданій комплексній точці.

4.48. Скласти програми обчислення суми всіх доданків, модуль яких не менше $\varepsilon > 0$, у комплексній точці z .

а) $\sin z = z - \frac{z^3}{3!} + \frac{z^5}{5!} - \dots + (-1)^n \frac{z^{2n+1}}{(2n+1)!} + \dots;$

б) $\cos z = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - \dots + (-1)^n \frac{z^{2n}}{(2n)!} + \dots;$

в) $sh z = z + \frac{z^3}{3!} + \frac{z^5}{5!} + \dots + \frac{z^{2n+1}}{(2n+1)!} + \dots;$

г) $ch z = 1 + \frac{z^2}{2!} + \frac{z^4}{4!} + \dots + \frac{z^{2n+1}}{(2n+1)!} + \dots;$

г) $e^z = 1 + \frac{z}{1!} + \frac{z^2}{2!} + \dots + \frac{z^n}{n!} + \dots;$

д) $\ln(1+z) = z - \frac{z^2}{2!} + \frac{z^3}{3!} - \dots + (-1)^n \frac{z^n}{n!} + \dots, \quad (|x| < 1);$

е) $\arctg z = z - \frac{z^3}{3!} + \frac{z^5}{5!} - \dots + (-1)^n \frac{z^{2n+1}}{2n+1} + \dots; \quad (|z| < 1).$

Тема 5 Символи та рядки

5.1. Скласти програму виведення великих літер латинського алфавіту в алфавітному порядку.

5.2. Визначити, чи є даний символ латинською літерою (великою або маленькою), цифрою або ні тим ні іншим.

5.3. Визначити, яка з двох заданих літер у даному рядку зустрічається частіше.

5.4. Визначити, чи входить до даного рядка кожна з літер слова *key*.

5.5. Перевірити, чи правильно в заданому рядку розставлені круглі дужки (тобто, чи знаходиться справа від кожної відкриваючої дужки відповідна їй закриваюча дужка, а зліва від кожної закриваючої - відповідна їй відкриваюча).

Відповідь - "так" або "ні".

5.6. Визначити, чи є заданий текст правильним записом цілого числа (можливо зі знаком).

5.7. Надрукувати заданий рядок:

а) виключивши з нього всі цифри і подвоївши знаки '+' та '-';

б) виключивши з нього всі знаки '+', безпосередньо за якими знаходиться цифра;

в) виключивши з нього всі літери 'e', безпосередньо перед якими знаходиться літера 'c';

г) замінивши в ньому всі пари 'ph' на літеру 'f';

г) виключивши з нього всі зайві пропуски, тобто з кількох, що йдуть підряд, залишити один.

5.8. Дано рядок, серед символів якого є принаймні одна кома. Знайти номер

а) першої по порядку коми;

б) останньої по порядку коми.

5.9. Виключити з заданого рядка групи символів, які знаходяться між '(' та ')'. Самі дужки теж мають бути виключені. Вважається, що дужки розставлено правильно (парами) та всередині кожної пари дужок немає інших дужок.

5.10. Заданий рядок, серед символів якого міститься двокрапка ':'. Отримати всі символи, розміщені

а) до першої двокрапки включно;

б) після першої двокрапки;

в) між першою і другою двокрапкою. Якщо другої двокрапки немає, то отримати всі символи, розміщені після єдиної двокрапки.

5.11. Заданий текст надрукувати по рядках, розуміючи під рядком або наступні 60 символів, якщо серед них немає коми, або частину тексту до коми включно.

5.12. Задана послідовність символів, яка має вигляд:

$$d_1 \pm d_2 \pm \dots \pm d_n$$

(d_i -цифри, $n > 1$), за якою знаходиться крапка. Обчислити значення цієї алгебраїчної суми.

5.13. Задане натуральне число n . Надрукувати в трійковій системі числення цілі числа від 0 до n .

5.14. У заданий рядок входять тільки цифри та літери. Визначити, чи задовольняє він наступній властивості:

а) рядок є десятковим записом числа, кратного 9 (6, 4);

б) рядок починається з деякої ненульової цифри, за якою знаходяться тільки літери і їх кількість дорівнює числовому значенню цієї цифри;

в) рядок містить (крім літер) тільки одну цифру, причому її числове значення дорівнює довжині рядка;

г) сума числових значень цифр, які входять в рядок, дорівнює довжині рядка;

д) рядок співпадає з початковим (кінцевим, будь-яким) відрізком ряду 0123456789;

е) рядок складається тільки з цифр, причому їх числові значення складають арифметичну прогресію (наприклад, 3 5 7 9, 8 5 2, 2).

5.15. Знайти у даному рядку символ та довжину найдовшої послідовності однакових символів, що йдуть підряд.

5.16. Скласти програму підрахунку загального числа входжень символів '+', '-', '*' у рядок A .

5.17. Скласти програму перетворення рядка A , замінивши у ньому всі знаки оклику '!' крапками '.', кожну крапку – трьома крапками '...', кожну зірочку '*' - знаком '+

5.18. Інверсія рядка A - це рядок B , записаний тими ж символами у зворотньому порядку. Інверсія порожнього рядка за означенням – порожній рядок. Побудувати інверсію рядка. Вирізки не використовувати.

5.19. Рядок називається симетричним, якщо його символи, рівновіддалені від початку та кінця рядка, співпадають.

Порожній рядок вважається симетричним. Перевірити рядок A на симетричність.

5.20. Скласти програму видалення із рядка A всіх входжень заданої групи символів.

5.21. Скласти програму перетворення слова A , видаливши у ньому кожний символ '*' та подвоївши кожний символ, відмінний від '*'.

5.22. Скласти програму підрахунку найбільшої кількості цифр, що йдуть підряд, у рядку A .

5.23. Скласти програму підрахунку числа входжень у рядок A заданої послідовності літер.

5.24. Скласти програму, яка за рядком A та символом S буде новий рядок, отриманий заміною кожного символу, слідуєчого за S , заданим символом C .

5.25. Скласти програму перетворення рядка A видаленням із нього всіх ком, які передують першій крапці, та заміною у ньому знаком '+' усіх цифр '3', які зустрічаються після першої крапки.

5.26. Скласти програму виведення на друк усіх цифр, які входять у заданий рядок, та окремо - решту символів, зберігаючи при цьому взаємне розташування символів у кожній з цих двох груп.

5.27. Рядок називається монотонним, якщо він складається з зростаючої або спадної послідовності символів. Скласти програму перевірки монотонності рядка.

5.28. Скласти програму обчислення числа входжень у рядок A символів, перелічених у рядку V .

5.29. Знайти символ, кількість входжень якого у рядок A

а) максимальна;

б) мінімальна.

5.30. Дано рядок A , що містить послідовність слів. Скласти програми, що визначають:

а) кількість усіх слів;

б) кількість слів, що починаються із заданого символу c ;

в) кількість слів, що закінчуються заданим символом c ;

г) кількість слів, що починаються й закінчуються заданим символом c ;

г) кількість слів, що починаються й закінчуються однаковим

символом.

5.31. Виділити з рядка A найбільший підрядок, перший і останній символи якого співпадають.

5.32. Перевірити, чи складається рядок з

а) 2 симетричних підрядків;

б) n симетричних підрядків.

5.33. Виділити з рядка найбільший монотонний підрядок, коди послідовних символів якого відрізняються на 1.

5.34. Замінити всі пари однакових символів рядка, які йдуть підряд, одним символом. Наприклад, рядок *'aabcbb'* перетворюється у *'abcb'*.

5.35. Побудувати рядок S з рядків $S1, S2$ так, щоб у S входили

а) ті символи $S1$, які не входять у $S2$;

а) всі символи $S1$, які не входять у $S2$, та всі символи $S2$, які не входять у $S1$.

5.36. Видалити з рядка симетричні початок та кінець. Наприклад, рядок *'abcdefba'* перетворюється у *'cdef'*.

5.37. Скласти програму виведення на друк тільки маленьких літер українського алфавіту, які входять в заданий рядок.

5.38. Заданий рядок, який складається з великих літер українського алфавіту. Скласти програму перевірки впорядкованості цих літер за алфавітом.

5.39. Скласти програму виведення на друк в алфавітному порядку усіх різних маленьких українських літер, які входять до даного рядка.

5.40. Написати програму, яка виконує зсув по ключу (ключ задається) тільки для малих латинських літер. Наприклад: вхідні дані *anz* – рядок, *2* – ключ. Результат: *срb*.

5.41. До голосних літер в латинському алфавіті відносяться літери A, E, I, O, U і Y . Інші літери вважаються приголосними. Напишіть програму, яка підраховує кількість голосних літер в тексті.

5.42. Рядок складається з маленьких латинських літер і проміжків. Подвойте в ньому всі голосні літери, тобто літери a, e, i, o, u та y .

5.43. Задано рядок, що містить електронну адресу користувача. Переконайтеся, що ця адреса є коректною

(наявність символу @ і крапки, а також наявність принаймні двох символів після останньої крапки).

5.44. Дано рядок, що містить букви та дужки кількох видів: '(', ')', '[', ']', '{', '}'. Перевірити, чи правильно в заданому виразі розставлені дужки.

5.45. Як відомо, римська система числення використовує символи M, D, C, L, X, V та I для чисел 1000, 500, 100, 50, 10, 5 та 1 відповідно. Інші числа записуються використовуючи різні комбінації вищенаведених символів, наприклад

I - 1	VII - 7	XLVI - 46	CCCII - 302
II - 2	VIII - 8	L - 50	CDXLI - 441
III - 3	IX - 9	LXXV - 75	ID - 499
IV - 4	X - 10	XCII - 92	D - 500
V - 5	XVIII - 18	IC - 99	DCXCV - 695
VI - 6	XXXI - 31	C - 100	CM - 900

Написати програму, що перетворює число записане у римській системі числення у десяткову і навпаки.

5.46. Дано рядок, що містить дріб чисельник і знаменник якого записані у римській системі числення. Необхідно скоротити цей дріб, записаний у римській системі числення.

5.47. Дано два натуральних числа записаних у римській системі числення. Знайти їхню суму, різницю та добуток. Результат має бути також записаним у римській системі числення.

5.48. Дано текст, що містить натуральні числа, операції відношень та логічні операції, таким чином, що цей текст є правильним записом деякого логічного виразу (умови). Наприклад:

$$1 > 3 \text{ and } 3 == 5$$

Обчислити значення цього логічного виразу.

Тема 6 Списки

.....

6.1. Скласти програму для обчислення суми:

а) компонент з парними номерами;

- б) компонент з непарними номерами;
- в) компонент, що діляться на 3 або 5;
- г) додатніх компонент;
- д) від'ємних компонент дійсного вектора.

6.2. Скласти програми для

- а) введення;
- б) виведення дійсного вектора.

6.3. Скласти програми обчислення

- а) середнього арифметичного компонент дійсного вектора;
- б) норми дійсного вектора;
- в) відстані між двома точками в n -вимірному евклідовому

просторі;

- г) скалярного добутку двох дійсних векторів.

6.4. Скласти програми обчислення кількості компонент дійсного вектора

- а) більших заданого числа;
- б) які належать заданому відрізьку прямої.

6.5. Визначити функцію обчислення вектора B за формулами

$$b_i = a_1 + a_2 + \dots + a_n, \quad i = 1, 2, \dots, n,$$

де a_i – компоненти заданого вектора A .

6.6. Дано масив, компонентами якого є коефіцієнти многочлена

$$P_n(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n.$$

Скласти програму обчислення:

- а) значення многочлена за заданого значення x ;
- б) похідної від многочлена за заданого значення x ;
- в) інтеграла від многочлена $P_n(x)$ на заданому відрізьку.

6.7. Скласти програму для обчислення суми й добутку двох многочленів

$$A(x) = a_0 + a_1x + \dots + a_nx^n \quad \text{і} \quad B(x) = b_0 + b_1x + \dots + b_mx^m.$$

6.8. Задано многочлен $P(x)$ з нульовим вільним членом. Знайти перші n коефіцієнтів розкладу $1/P(x)$ в ряд Тейлора в нулі.

6.9. Скласти підпрограму обчислення частки $Q(x)$ і залишку $R(x)$ від ділення многочлена $P(x) = p_0 + p_1x + \dots + p_nx^n$ на многочлен $T(x) = t_0 + t_1x + \dots + t_mx^m$ за схемою $P(x) = T(x) \cdot Q(x) + R(x)$, $\text{deg}(R) < \text{deg}(T)$, де deg – степінь многочлена.

6.10. Написати програму для обчислення найбільшого спільного дільника двох многочленів.

6.11. Задано два вектори $X = \{x_i\}$, $Y = \{y_i\}$, $i = 0, \dots, n$, причому серед x_i немає однакових. Знайти в точці t значення інтерполяційного полінома:

а) у формі Ньютона (реалізувати програму без використання додаткових масивів):

$$P_n(t) = x_0 + (t - x_0)Y(x_0; x_1) + (t - x_0)(t - x_1)Y(x_0; x_1; x_2) + \dots + (t - x_0)(t - x_1) \dots (t - x_{n-1})Y(x_0; x_1; x_2; \dots; x_n),$$

де введено позначення

$$Y(x_i; x_j) = \frac{y_i - y_j}{x_i - x_j}; \quad Y(x_i; x_j; x_k) = \frac{Y(x_i; x_j) - Y(x_j; x_k)}{x_i - x_k}; \quad \dots$$

б) у формі Лагранжа:

$$P_n(t) = y_0 L_0(t) + y_1 L_1(t) + \dots + y_n L_n(t),$$

де введено позначення

$$L_k(t) = \frac{W_{n+1}(t)}{(t - x_k)W'_{n+1}(x_k)}; \quad W_{n+1}(t) = (t - x_0)(t - x_1) \dots (t - x_n).$$

6.12. Слід квадратної матриці - це сума її діагональних елементів. Скласти програму обчислення сліду матриці.

6.13. Скласти програму для обчислення

а) визначника квадратної матриці;

б) ранга квадратної матриці.

6.14. Скласти програми для обчислення

а) суми усіх недиагональних елементів матриці;

б) кількості нульових елементів матриці.

6.15. Скласти програми для обчислення:

а) суми двох векторів;

б) добутку вектора на число.

6.16. Розглядаючи вектори A та B як послідовності цифр десяткового запису деяких невід'ємних цілих чисел, отримати вектор C - аналогічне представлення для суми цих двох чисел.

6.17. Циклічним k -зсувом вектора (a_1, a_2, \dots, a_n) вліво називається вектор $(a_{k+1}, \dots, a_n, a_1, \dots, a_k)$. Скласти програму для його обчислення.

6.18. Визначити циклічний k -зсув вправо та програму для його обчислення.

6.19. Скласти програми

- а) обміну значень двох векторів;
- б) перестановки компонент вектора в зворотньому порядку.

6.20. Скласти програму пошуку

- а) максимальної; б) мінімальної компоненти вектора.

6.21. Скласти програму пошуку в заданому векторі компоненти:

- а) більшої за задане число;
- б) такої, що належить заданому відріzkу на прямій.

6.22. Скласти програму підрахунку кількості максимальних і мінімальних компонент вектора.

6.23. Скласти програму одночасного обчислення максимальної і мінімальної серед компонент з парними і непарними номерами.

6.24. Скласти програму для обчислення суми компонент дійсного вектора, які розташовані між максимальною та мінімальною компонентами (всі компоненти вектора різні).

6.25. Задані координати n точок на площині $(x_1, y_1), \dots, (x_n, y_n)$. Знайти номери двох точок, відстань між якими найбільша (вважати, що така пара точок єдина).

6.26. Дано два дійсних вектори довжини n . Скласти програму пошуку найменшої серед тих компонент першого вектора, які входять до другого вектора.

6.27. Скласти програму пошуку спільної компоненти двох векторів.

6.28. Скласти програму пошуку в заданому векторі

- а) двох компонент, що дорівнюють заданому числу;
- б) двох нульових компонент, які йдуть підряд;
- в) двох рівних компонент;
- г) двох рівних компонент, які йдуть підряд;
- д) компонент, що є числами Фібоначі.

6.29. Скласти програму обчислення кількості інверсій в заданому векторі (тобто таких пар компонент, в яких більше число знаходиться зліва від меншого: $x_i > x_j$ при $i < j$).

6.30. Скласти програму, яка перевіряє впорядкованість вектора за зростанням. (Вектор (a_1, a_2, \dots, a_n) впорядкований за зростанням, якщо $a_1 < a_2 < \dots < a_n$).

6.31. Скласти програму пошуку спільної компоненти двох впорядкованих векторів.

6.32. Скласти програму перетворення дійсного вектора за наступним правилом: всі від'ємні компоненти вектора перенести в його початок, а всі інші - в кінець, зберігаючи початкове взаємне розташування як серед від'ємних, так і серед інших компонент.

6.33. Визначити два впорядкованих за неспаданням вектори. Скласти програму об'єднання

- а) всіх компонент;
- б) спільних компонент

цих двох векторів у третій, щоб він знову став впорядкованим за неспаданням.

6.34. Задана дійсна матриця розміру $m \times n$. Знайти вектор B , k -та компонента якого b_k це:

- а) сума абсолютних величин елементів k -го рядка матриці;
- б) добутку елементів k -го рядка;
- в) значення середнього арифметичного елементів k -го рядка;
- г) число від'ємних елементів в k -му рядку;
- д) добутку квадратів тих елементів k -го рядка, модулі яких належать відріzkу $[1; 1,5]$ (якщо таких елементів немає, то покласти $b_k = 1$);
- е) значення першого по порядку додатнього елемента k -го рядка (якщо таких елементів немає, то покласти $b_k = 10$);
- є) сума елементів, які знаходяться за першим від'ємним елементом в k -му рядку (якщо таких елементів немає, то покласти $b_k = 100$).

6.35. Написати програми для:

- а) транспонування матриці;
- б) множення матриці на вектор;
- в) перестановки двох заданих рядків (стовпчиків) матриці;
- г) перестановки заданого рядка матриці з заданим її стовпчиком;

- д) побудови цілочисельної квадратної матриці порядку 7, елементами якої є числа 1, 2, ..., 49, розташовані в ній по спіралі;
 е) видалення із матриці заданого рядка і заданого стовпчика.

6.36. Задано дійсну квадратну матрицю порядку $2n$. Побудувати нову матрицю, переставляючи її блоки розміру $n \times n$:
 а) відповідно до рис. 6.3 а; б) відповідно до рис. 6.3 б.

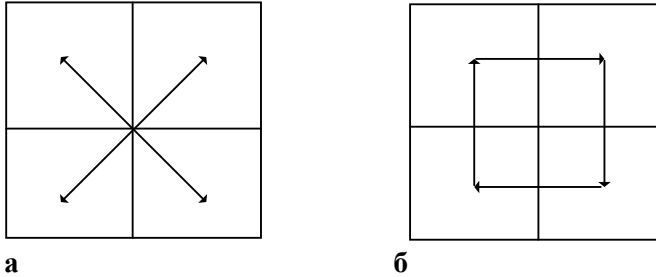


Рис. 6.3

6.37. Задані натуральне число m , цілі числа a_1, a_2, \dots, a_m і цілочисельна квадратна матриця порядку m . Рядок з номером i матриці назвемо відміченим, якщо $a_i > 0$, і невідміченим у протилежному випадку.

а) всі елементи, розташовані у відмічених рядках матриці, перетворити за правилом: додатні замінити на -1, від'ємні - на 1, а нульові залишити без зміни;

б) підрахувати кількість від'ємних елементів матриці, розташованих у відмічених рядках.

6.38. Задано дійсні матриці A, B, C, N порядку n , причому N - нульова матриця. Побудувати за цими матрицями матрицю D розміру $2n \times 3n$ у вигляді

$$D = \begin{pmatrix} A & B & C \\ B & N & A \end{pmatrix}.$$

6.39. Скласти програму пошуку заданого елемента матриці.

6.40. Задана цілочисельна квадратна матриця порядку n і цілочисельний вектор довжини m . Замінити нулями в матриці ті елементи, для яких існують рівні серед компонент даного вектора.

6.41. Скласти програму обчислення норм дійсної матриці $A = (a_{ij})_{i,j=1,\dots,n}$:

а) $\|A\| = \max_{i,j=1,\dots,n} |a_{ij}|$;

б) $\|A\| = \sum_{i,j=1}^n |a_{ij}|$;

в) $\|A\| = \max_{i=1,\dots,n} \sum_{j=1}^n |a_{ij}|$;

г) $\|A\| = \max_{j=1,\dots,n} \sum_{i=1}^n |a_{ij}|$;

ґ) $\|A\| = \sqrt{\sum_{i,j=1}^n a_{ij}^2}$;

д) $\|A\| = \sqrt[p]{\sum_{i,j=1}^n a_{ij}^p}$.

6.42. Скласти програму обчислення у матриці

а) найменшого елемента;

б) найбільшого елемента;

в) суми елементів рядка, в якому розташований елемент з найменшим значенням;

г) суми найбільших значень елементів її рядків;

д) суми елементів рядків з від'ємним елементом головної діагоналі.

6.43. Скласти програму перестановки місцями рядка матриці, що містить елемент з найменшим значенням із рядком, в якому міститься елемент з найменшим значенням.

6.44. Скласти програму пошуку найменшого серед найбільших елементів рядків квадратної дійсної матриці порядку n , тобто $\|A\| = \min_{1 \leq i \leq n} \max_{1 \leq j \leq n} \{a_{ij}\}$.

6.45. Скласти програму пошуку в матриці

а) індексів усіх її ненульових елементів;

б) кількості усіх її різних елементів.

6.46. Дана дійсна матриця розміру $m \times n$. Скласти програму побудови вектора, компонентами якого є

а) найменше значення елементів рядків;

б) різниці між найбільшим і найменшим значеннями елементів рядків;

в) найбільші із значень елементів рядків від початку до головної діагоналі включно.

6.47. Елемент матриці називається "особливим", якщо:

1) він більший за суму інших елементів свого стовпчика;

2) в його рядку зліва від нього знаходяться елементи, менші за нього, а справа - більші.

Скласти програму пошуку числа "особливих" елементів матриці.

6.48. Елемент матриці називається слабким локальним максимумом, якщо з нього не можна зсунутись на крок в жодному з чотирьох напрямів в бік більшого елемента. Елемент називається сильним локальним максимумом, якщо з нього не можна зсунутись по жодному з восьми напрямків.

Скласти програму обчислення кількості сильних і слабких максимумів матриці.

6.49. Скласти програму перевірки матриці на симетричність.

6.50. Скласти програму побудови за заданою матрицею A цілочисельного вектора b , присвоївши його k -ій компоненті значення 1, якщо виконується вказана нижче умова, і значення 0 в протележному випадку:

а) всі елементи k -го стовпчика однакові;

б) елементи k -го рядка матриці впорядковані за спаданням;

в) k -ий рядок матриці симетричний;

г) елемент k -го рядка, що належить головній діагоналі, симетричний;

д) елементи k -го рядка не перебільшують заданого числа x ;

е) в k -му рядку матриці є хоча б один від'ємний елемент.

6.51. Скласти програму перестановки рядків і стовпчиків матриці так, щоб найбільший елемент її опинився у верхньому лівому куті.

Вказівка. Знаходимо номер рядка k і номер стовпчика l , що містять найбільший елемент матриці. Потім міняємо місцями 1-ий і k -ий рядки, 1-ий і l -ий стовпчики матриці.

6.52. Елемент матриці назвемо сідловою точкою, якщо він є найменшим в своєму рядку і водночас найбільшим в своєму стовпчику. Для заданої цілочисельної матриці скласти програму пошуку індексів усіх її сідлових точок.

6.53. Дана дійсна квадратна матриця, всі елементи якої різні. Скласти програму обчислення скалярного добутку рядка, в якому знаходиться найбільший елемент матриці, на стовпчик з найменшим елементом.

6.54. Скласти програму, яка перевіряє, чи є задана цілочисельна квадратна матриця ортонормованою, тобто. такою, в якій скалярний добуток кожної пари різних рядків дорівнює 0, а скалярний добуток кожного рядка на себе дорівнює 1.

6.55. Скласти програму, що перевіряє чи є задана цілочисельна квадратна матриця магічним квадратом, тобто такою, в якій суми елементів в усіх рядках і стовпчиках однакові.

6.56. Дана бульова матриця. Відомо, що множина істинних елементів є об'єднанням прямокутників, що не перетинаються. Скласти програму обчислення їх кількості.

6.57. В умовах попереднього завдання скласти програму підрахунку кількості прямокутників, що є квадратами.

6.58. Задане деяке розташування чорних і білих пішаків (в довільній кількості) на шаховій дошці. Скласти програму підрахунку

- а) кількості можливих ходів білими;
- б) кількості взятъ білими;
- в) кількості чорних і білих пішаків під боєм.

6.59. Таблиця футбольного чемпіонату задана квадратною матрицею порядку n , в якій всі елементи, що належать головній діагоналі, дорівнюють нулеві, а кожний елемент, що не належить головній діагоналі, дорівнює 3, 1 або 0 (числу очок, набраних у грі: 3 - вигреш, 1-нічия, 0-прогреш). Скласти програму

- а) знаходження кількості команд, що мають перемог більше, ніж поразок;
- б) визначення номерів, що пройшли чемпіонат без поразок;
- в) пошуку команд, що виграли більше половини ігор.

6.60. Задана бульова матриця СУСІДИ, елементи якої СУСІДИ(a,b) рівні істині, якщо країни a і b мають спільний кордон, і хибності - в протилежному випадку. Скласти процедуру пошуку країни, що має найбільшу кількість сусідів серед перелічених країн.

6.61. Задана дійсна матриця T , елементи якої $T(x, y)$ означають середньомісячну температуру на острові x в місяці y . Написати програму, що визначає який місяць і на якому острові серед перелічених островів є найхолоднішим.

6.62. Задана непротирічна таблиця спорідненості ТС, елементи якої $ТС(x, y)$ можуть приймати значення із списку (*син, дочка, батько, мати, ні*). Наприклад, $ТС(x, y)=ні$, якщо y не є ні одним з батьків, ні родичем, ні дитиною x'_a , $ТС(x, y)=син$, якщо y - син x'_a і т.п. Скласти програму для пошуку

- а) будь-якої з онучок;
- б) будь-якого з дядьків;
- в) кількості двоюрідних братів і сестер людини з ім'ям n із заданого списку.

6.63. Задана непорожня послідовність непорожніх слів з латинських літер. Словами називаються групи символів, які розділені одним чи кількома пропусками та не містять пропусків всередині себе. Визначити кількість слів, які:

- а) містяться в даній послідовності;
- б) починаються з заданої літери;
- в) закінчуються заданою літерою;
- г) починаються і закінчуються однією літерою;
- д) містять принаймні одну задану літеру;
- е) містять рівно три заданих літери.

6.64. В умовах попереднього завдання:

- а) знайти довжину найкоротшого слова;
- б) підрахувати кількість входжень заданої літери в останнє слово даної послідовності.

6.65. Видалити з рядка всі повторні входження символів.

6.66. Словами називаються підрядки, які розділені одним чи кількома пропусками та не містять пропусків всередині себе. За рядком A Скласти програму виведення на друк:

- а) всіх слів рядка у зворотньому порядку;
- б) усіх слів, які зустрічаються у рядку по одному разу;
- в) цього ж рядка, але з видаленням з нього повторних входжень слів.

6.67. Знайти

- а) найкоротше слово рядка;
- б) найдовше слово рядка.

6.68. Показати усі слова рядка у порядку неспадання їх довжини.

6.69. Дано два впорядкованих списки. Не використовуючи

функцію сортування, об'єднати ці списки у один впорядкований список.

6.70. Задано номер мобільного телефону. Визначити, які цифри відсутні в цьому номері.

6.71. Задано два натуральних числа n і m . Вивести матрицю, що складається з n рядків та m стовпчиків, заповнену натуральними числами від 1 до $n * m$.

6.72. Розв'язати систему двох лінійних рівнянь з двома невідомими за формулами Крамерами. Система рівнянь, наведена у вхідних даних, має вигляд:

$$\begin{cases} 5x + 8y = 11 \\ -3x + 6y = 15 \end{cases}$$

Відомо, що система рівнянь завжди має єдиний розв'язок.

Тема 7 Кортєжі

.....

7.1. Гральні карти мають 2 поля:

*масть - піки, хрести, бубни, чирви та
значення - шість, сім, вісім, дев'ять, десять, валет, дама,
король, туз;*

Скласти програму, яка перевіряє, чи "б'є" карта $K1$ карту $K2$ враховуючи те, що масть KM є козирною .

7.2. Визначивши поле шахової дошки, скласти програму, яка перевіряє, чи може ферзь за один хід перейти з поля $h1$ шахової дошки на поле $h2$.

7.3. Заданий вектор розмірності n , компонентами якого є записи, що містять відомості про вершини гір. У відомостях про кожну вершину вказуються назва гори і її висота. Скласти програму пошуку найвищої вершини.

7.4. Заданий вектор ГР, компонентами якого $ГР(x)$ є записи, що містять дані про людину з іменем X із вказаного списку. Кожне

дане складається з вказаної статі людини та її зросту. Скласти програми для

- а) обчислення середнього зросту жінок з групи ГР;
- б) пошуку найвищого чоловіка;
- в) перевірки, чи є в групі ГР дві людини, однакові на зріст.

7.5. Заданий вектор ГР розмірності n , компонентами якого є записи, що містять анкети службовців деякого закладу. В кожній анкеті вказується прізвище службовця, його стать, дата народження у вигляді: число, місяць, рік. Скласти програму пошуку

- а) найстаршого з чоловіків групи ГР;
- б) людей з групи ГР, прізвища яких починаються з заданої літери.

7.6. Заданий вектор розмірності n , компоненти якого містять інформацію про студентів деякого вузу. Відомості про кожного студента складаються з вказання його прізвища, ім'я, по батькові, статі, віку, курсу. Скласти програму пошуку

- а) найбільш поширених чоловічих та жіночих імен;
- б) прізвищ та ініціалів усіх студентів, вік яких є найбільш поширеним.

7.7. Заданий вектор розмірності n , компонентами якого є відомості про складання іспитів студентами деякого вузу. Інформація про кожного студента задана в наступному вигляді: прізвище, номер групи, оцінка_1, оцінка_2, оцінка_3. Скласти програму пошуку

- а) студентів, що мають заборгованості хоча б з одного з предметів;
- б) предмета, який було здано краще за усі інші;
- в) студентів, що склали всі іспити на 5 і 4.

7.8. Задано вектор C розміру n , компонентами якого є відомості про мешканців деяких міст. Інформація про кожного мешканця містить його прізвище, назву міста мешкання; адреси мешкання у вигляді: вулиця, будинок, квартира. Скласти програму пошуку двох будь-яких жителів із списку C , що мешкають в різних містах за однаковою адресою.

7.9. Задано вектор Рейс, компонентами якого Рейс(x) є множини міст із вказаного списку, в які можна за один рейс

доїхати автобусом з міста x . Скласти програму пошуку множини міст, в які можна дістатися автобусом (за один рейс або через інші міста) із заданого міста.

7.10. Задані n точок площини. Провести коло, на якому лежить найбільша кількість точок даної множини.

7.11. Задані n точок площини. Скласти програму підрахунку кількості рівнобічних трикутників з вершинами у цих точках.

7.12. Задані n точок площини. Скласти програму вибору трьох з них таких, щоб периметр трикутника з вершинами в цих точках був найбільшим.

7.13. Скласти програму пошуку найменшого серед найбільших елементів рядків квадратної дійсної матриці порядку n , тобто $\|A\| = \min_{1 \leq i \leq n} \max_{1 \leq j \leq n} \{a_{ij}\}$.

Вказівка: використати функції zip та map.

7.14. Скласти програму перевірки квадратної дійсної матриці порядку n , на симетричність

Вказівка: використати функції zip та map та транспоновану матрицю.

7.15. Скласти програму для обчислення суми всіх елементів дійсної матриці.

Вказівка: використати функції map та sum.

7.16. Скласти програму для обчислення суми всіх діагональних елементів дійсної квадратної матриці.

Вказівка: використати функцію sum та спискоутворення.

Тема 8 Словники

.....

8.1. Відомості про учня складаються з його унікального номера учнівського квитка, імені, прізвища та назви класу (рік навчання та літери), в якому він вчиться. Дано словник, який містить відомості про учнів школи. Скласти програми, які дозволяють

а) визначити, чи є в школі учні з однаковим прізвищем;

б) визначити, чи є учні з однаковим прізвищем в якихось паралельних класах;

в) визначити, чи є учні з однаковим прізвищем у якомусь класі;

г) відповісти на питання а)-в) по відношенню до учнів, у яких співпадають і ім'я, і прізвище;

д) визначити, в яких класах нараховується більше, ніж 35 учнів;

е) визначити, на скільки учнів в восьми класах більше, ніж в десятих;

є) зібрати у словник відомості про учнів 9-х і 10-х класів, помістивши спочатку відомості про учнів класу 9а, потім 9б і т.д.

ж) отримати список учнів даного класу за наступними зразками:

прізвище ім'я

прізвище і.

і.прізвище.

8.2. Дано словник, який містить, ті ж самі відомості про учнів школи, що і в попередній задачі, та додатково оцінки, отримані учнями на екзаменах за заданими предметами. Скласти програми для

а) визначення кількості учнів, які не мають оцінок нижче чотирьох;

б) побудови списку, який містить відомості про кращих учнів школи, тобто про учнів, які мають оцінки не нижче чотирьох;

в) друку відомостей про учнів, які мають хоча б одну незадовільну оцінку, у вигляді: прізвище та ініціали, назва класу, предмет.

8.3. Відомості про автомобіль складаються з його марки, унікального номера і прізвища власника. Дано словник, який містить відомості про декілька автомобілей. Скласти процедури знаходження

а) прізвищ власників номерів автомобілей даної марки;

б) кількості автомобілей кожної марки.

8.4. У двох списках міститься таблиця футбольного турніру. У першому списку записано назви команд. У другому списку – результати матчів у вигляді кортежу:

(команда1, команда2, m1, m2)

де, команда1, команда2 – перша та друга команда;

m1, m2 – кількість м'ячів, забитих відповідно першою та другою командою.

Показати команду, яка є лідером, якщо за перемогу нараховується 3 очки, за нічию – 1, за поразку – 0.

З двох команд, які мають однакову кількість очок, першою вважається та, яка

1) має кращу різницю забитих та пропущених м'ячів;

2) при однаковій різниці має більше забитих м'ячів;

3) при всіх однакових попередніх показниках визначається жеребкуванням (використати для жеребкування генератор випадкових чисел).

Показати поточну таблицю турніру у вигляді:

місце, команда, ігор, вигравів, нічиїх, поразок, м'ячів забито, м'ячів пропущено, очок

Вказівка. Створити словник таблиці турніру з ключем Команда та кортежем.

8.5. У словнику для англійських слів дано їх переклади українською. Причому, для одного слова може бути декілька перекладів – синонімів. Скласти програму, яка за даним словником буде інший словник, у якому українські слова є ключам і для кожного даються переклади англійською.

8.6. Ненульові елементи розрідженої матриці зберігаються у словнику. Ключами словника є кортежі, що складаються з індексів рядка та стовпчика, а значеннями словника, - значення елементів матриці. Обчислити:

а) суму двох розріджених матриць однакового розміру

б) добуток двох розріджених матриць

в) значення мінімального елемента розрідженої матриці

г) значення максимального елемента розрідженої матриці

д) добуток розрідженої матриці на розріджений вектор (зберігається також у словнику)

е) чи є розріджена матриця нижньою трикутною

є) чи є розріджена матриця діагональною

ж) транспоновану розріджену матрицю

8.7. Нехай є бінарне зображення (складається з білих та чорних точок). Частина цього зображення задаються у так званому «ланцюговому коді», у якому по рядках виділяються області, зафарбовані чорним. У цьому коді позначається: початок зафарбованої ділянки, кількість чорних точок, що йдуть підряд. Обчислити кількість фігур, утворених у зображенні.

8.8. Скласти програму, яка перевіряє впорядкованість вектора за зростанням з використанням функції all(). (Вектор (a_1, a_2, \dots, a_n) впорядкований за зростанням, якщо $a_1 < a_2 < \dots < a_n$).

8.9. Скласти програму обчислення кількості інверсій в заданому векторі (тобто таких пар компонент, в яких більше число знаходиться зліва від меншого: $x_i > x_j$ при $i < j$) з використанням функції filter().

8.10. Заданий список, у якому містяться відомості щодо постачання товарів з України до інших країн. Список складається з кортежів:

(країна, товар, кількість)

З'ясувати

А) Усі товари, що постачаються до певної країни а також їх кількість.

Б) Усі країни, до яких постачається заданий товар

Вказівка: побудувати словник з ключем «країна». Значеннями у словнику мають бути словники з ключем «товар» та значенням – кількість.

8.11. Визначити, кількість різних символів у рядку.

8.12. Знайти символ, який входить у текст найбільшу кількість разів.

8.13. В Інституті філології університету кожен зі студентів вивчає принаймні чотири іноземних мови. Дано словник студентів інституту філології у якому у якості значення міститься список мов, які вивчає студент. Визначте:

а) всі мови, що вивчаються у інституті філології;

б) мови, які вивчають всі студенти інституту;

в) мови, які вивчає переважна більшість;

г) мови, які вивчає менше чверті студентів.

8.14. Задано рядок, що складається з латинських (українських)

літер. Надрукувати всі літери, що

- а) входять до рядка по одному разу;
- б) входять до рядка не менше двох раз;
- в) не входять до рядка.

8.15. Дано список країн та міст, що знаходяться у кожній з заданих країн. Для заданого міста знайдіть країну у якій воно знаходиться.

8.16. Нехай словник моделює стан гаманця у такий спосіб: значення банкноти (номінал) – це ключ, а кількість банкнот цього номіналу – значення.

якщо у гаманці відсутня банкнота певного номіналу, то такий елемент відсутній у гаманці (не допускається у словнику наявність пар типу {20:0})

Потрібно порахувати суму коштів, що міститься у гаманці. Вважається, що повний перелік номіналів банкнот відомий.

8.17. Відомі дані про масу й об'єм N предметів, виготовлених із різних матеріалів. Знайти предмет, густина матеріалу якого найбільша.

8.18. Використовуючи словники запропонувати та реалізувати метод шифрування заданого рядка символів та дешифрування його до вихідного рядка.

8.19. Відомі дані про чисельність населення (у мільйонах жителів) та площі N держав. Знайти країну з мінімальною щільністю населення.

8.20. Дано словник, що у якості ключів містить назви пір року (зима, весна, літо, осінь), а у ролі значень – кортеж з назв місяців року, що належать до цієї пори року. Здано таблицю, що задається словником з ключами номер місяця у році, і значеннями – показниками середньомісячних опадів у деякому регіоні. Знайти:

- а) пору року на яку припадає найбільша кількість опадів;
- б) найсухішу пору року;
- в) пору року з найбільшими середньомісячними коливаннями кількості опадів;

8.21. Чи можна з літер слова a скласти слово b , причому кожному літеру можна використати тільки один раз?

8.22. Відредагувати задане речення, видаливши з нього ті слова, які зустрічаються в реченні задану кількість разів.

8.23. На площині задано N точок. Вивести кути нахилу та вільні члени всіх прямих, які проходять більш ніж через одну точку із заданих. Для кожної прямої вказати, через скільки точок вона проходить. Використовувати словник.

8.24. На площині задано N відрізків у вигляді двох точок, кожна з яких задана парою дійсних координат. Використовуючи словник, знайти точку перетину двох відрізків, що має мінімальну абсцису.

8.25. Нехай значення функції $f(n)$ – кількість літер у письмовому представленні числа n ($f(1)=4$ (“один”), $f(3)=3$ (“три”), $f(42)=8$ (“сорок два”), $a(2001)=13$ (“дві тисячі один”)). Знайдіть всі числа до 10000, для яких $f(n)=n$. (Вказівка: Використовуйте словник для зберігання кількості літер у представленні цифри)

8.26. Введений текст T . Підрахувати число повторень в ньому малих латинських літер ('a' - 'z') і вивести текст рядки якого мають вигляд: “<літера> - <кількість повторень даної літери>”. Літери, відсутні в тексті, не виводити. Рядки ці потрібно впорядкувати за спаданням кількості повторень літер, а при однаковій кількості повторень – по зростанню кодів літер.

8.27. Вводиться список цілих чисел V . У кожній підмножині його елементів, що мають однакову початкову цифру, знайдіть суму значень цих елементів, за виключенням початкового елемента групи (передбачається, що елементи групи розміщуються в тому ж порядку, що і у вихідному векторі). Якщо група складається з єдиного елемента, то сума повинна дорівнювати 0. Для кожної групи потрібно обчислити відповідну їй цифру та знайдену суму.

8.28. Вводиться список відомостей про клієнтів фітнес-центру. Кожний елемент послідовності включає наступні цілі поля: <Код клієнта> <Рік> <Номер місяця> <Кількість занять (у годинах)> Знайти елемент послідовності з мінімальною тривалістю занять. Виведіть цю інформацію, а також відповідні їй рік і номер місяця (у вказаному порядку в тому ж рядку). Якщо таких елементів декілька — виведіть усі в довільному порядку.

8.29. Вводиться список кортежів в яких записано вартість, вік та модель легкових автомобілів. Визначити середню вартість автомобілів, вік яких більший за 5 років.

8.30. Вводиться список іменованих кортежів, де записано інформацію про ціну, наклад та назву журналу. Знайти середню вартість журналів, наклад яких менший за 1000 примірників.

8.31. У циклі, що завершується введенням порожнього рядку вводяться рядки, де через кому записано дані про масу, об'єм предметів, та матеріал, з якого зроблений предмет. Знайти предмет, густина матеріалу якого найбільша.

8.32. У циклі вводяться дані про чисельність населення (у мільйонах жителів) та площі N держав. Знайти країну з мінімальною щільністю населення.

8.33. Вводиться список відомості про оцінки учнів за трьом предметами: алгебра, геометрія та інформатика. Кожний елемент послідовності містить дані про одну оцінку та має наступні поля: <Прізвище> <Ініціали> <Клас> <Назва предмета> <Оцінка>. Повних однофамільців (із спільною фамілією та ініціалами) серед учнів немає. Клас задається цілим числом, оцінка - цілим числом в діапазоні 1–12. Назва предмета вказується в верхньому регістрі. Для кожного учня визначити середню оцінку по кожному з предметів і вивести її з двома знаками після крапки (якщо не має жодної оцінки, то вивести для цього предмету 0,00). Відомості про кожного учня виводити на окремому рядку, вказавши прізвище, ініціали та середні оцінки по алгебрі, геометрії та інформатиці.

Тема 9 Підпрограми

.....

9.1. Скласти програму обчислення добутку

$$p = f_0 * f_1 * \dots * f_n,$$

$$\text{де } f_l = \frac{1}{l^2 + 1} + \frac{1}{l^2 + 2} + \dots + \frac{1}{l^2 + l + 1}.$$

9.2. Два простих числа називаються "близнюками", якщо вони відрізняються один від одного на 2 (наприклад, числа 41 та 43). Скласти програму виведення на друк всіх пар "близнюків" з відрізка $[n, 2*n]$, де n - задане ціле число, яке більше 2.

9.3. Дано натуральне число n та послідовність натуральних чисел a_1, a_2, \dots, a_n . Показати всі елементи послідовності, які є

- а) повними квадратами;
- б) степенями п'ятірки;
- в) простими числами.

Визначити відповідні функції для перевірки, чи є число: повним квадратом, степенню п'ятірки, простим числом.

9.4. Дано натуральне число n . Для чисел від 1 до n визначити всі такі, які можна представити у вигляді суми двох повних квадратів. Описати функцію, яка перевіряє, чи є число повним квадратом.

9.5. Дано парне число $n > 2$. Перевірити для нього гіпотезу Гольдбаха, яка полягає в тому, що кожне парне число $n > 2$ можна представити у вигляді суми двох простих чисел. Визначити функцію, яка перевіряє, чи є число простим.

9.6. Скласти програму обчислення величини

$$\frac{\sqrt[3]{a - \sqrt{a^2 + 1}}}{1 + \sqrt[7]{3 + a}}$$

для заданого дійсного числа $a > 0$. Визначити функцію обчислення коренів $y = \sqrt[k]{x}$ з точністю ε за наступною ітераційною схемою

$$y_0 = 1; \quad y_{n+1} = y_n + (x / y_n^{k-1} - y_n) / k, \quad (n = 0, 1, 2, \dots),$$

взявши за відповідь наближення y_{n+1} , для якого $|y_{n+1} - y_n| < \varepsilon$.

9.7. Використовуючи функцію $y = \arctg(x)$ (*math.atan(x)*), $y = \arctg(x)$ (*math.atan(x)*), скласти підпрограму для обчислення функції, заданої співвідношенням

$$\operatorname{arctg}(x, y) = \begin{cases} \operatorname{arctg} \frac{x}{y}, & \text{якщо } y > 0; \\ \frac{\pi}{2}, & \text{якщо } x > 0 \text{ і } y > 0; \\ \pi + \operatorname{arctg} \frac{x}{y}, & \text{якщо } x \geq 0 \text{ і } y < 0; \\ -\frac{\pi}{2}, & \text{якщо } x < 0 \text{ і } y = 0; \\ -\pi + \operatorname{arctg} \frac{x}{y}, & \text{якщо } x < 0 \text{ і } y < 0. \end{cases}$$

9.8. Скласти програму обчислення значень функції $f(x)$, періодичної з періодом 1 і визначеної на всій числовій вісі. Графік функції зображено на рисунку 9.1

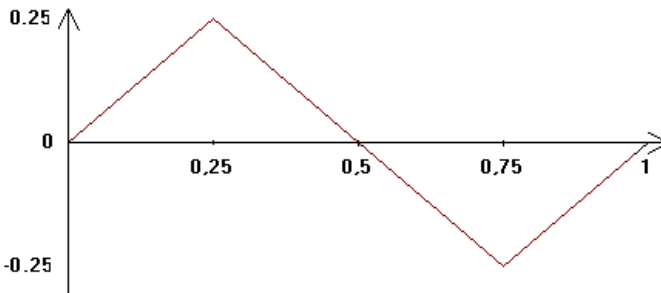


Рис. 9.1

Які допоміжні підпрограми будуть потрібні для розв'язку задачі ?

9.9. Визначити функцію для обчислення еліптичного інтегралу

$$I = \int_a^b \frac{dx}{a^2 \cos^2 x + b^2 \sin^2 x}, \quad (a < b),$$

який рівний границі $I = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n$ монотонно-збіжних послідовностей a_n і b_n , що визначаються рекурентними співвідношеннями

$$a_0 = 0; b_0 = b; a_{n+1} = \sqrt{a_n b_n}; b_{n+1} = \frac{a_n + b_n}{2}, \quad n = 1, 2, \dots$$

Вказана границя називається арифметико-геометричним середнім чисел a і b .

Вказівка. При виборі умови продовження циклу врахувати, що

$$a = a_0 < a_1 < \dots < a_n < \dots < b_n < \dots < b_0 = b.$$

9.10. Визначити функції для обчислення

а) синуса ; б) косинуса

використовуючи їх розклади в ряд Тейлора.

9.11. Дано координати вершин трикутника і точки всередині його. Використовуючи функцію для обчислення площі трикутника через три його сторони, визначити відстань від даної точки до найближчої сторони трикутника.

Вказівка. Врахувати що площа трикутника обчислюється також через основу і висоту.

9.12. Скласти функцію перевірки заданого рядка на симетричність.

9.13. Перевірити, чи є даний рядок ідентифікатором, натуральним числом, чи ні тим ні іншим. Скласти функції, які визначають чи є заданий символ літерою та чи є даний символ цифрою.

9.14. Скласти функцію, яка визначає позицію першого (останнього) входження заданого символу в заданий рядок.

9.15. Скласти процедуру, яка замінює в початковому рядку символів всі одиниці на нулі, а всі нулі - на одиниці. Заміна повинна виконуватись, починаючи з заданої позиції рядка.

9.16. Скласти процедуру, в результаті звернення до якої з першого заданого рядка видаляється кожний символ, який належить і другому заданому рядку.

9.17. Скласти підпрограму для обчислення значення натурального числа за заданим рядком символів, який є записом цього числа у системі числення за основою b ($2 \leq b \leq 16$).

Використати функцію, яка за заданим символом повертає відповідну цифру у системі числення за основою b .

9.18. Скласти підпрограму для отримання за заданим натуральним числом рядка символів, який є записом цього числа у системі числення за основою b ($2 \leq b \leq 16$). Використати функцію, яка за заданою цифрою у системі числення за основою b повертає символ, що відповідає цій цифрі.

9.19. Скласти алгоритм додавання „у стовпчик” двох чисел, записаних у вигляді рядків, що є позиційними записами цих чисел у десятковій системі числення. Використати підпрограми:

- 1) функцію *get_digit(c)* отримання цифри за символом c ;
- 2) функцію *get_symbol(d)* отримання символу за цифрою d ;
- 3) функцію *add_digit(n1, n2, p, n)* додавання двох цифр $n1, n2$ з урахуванням перенесення p та отримання останньої цифри результату n ;

- 4) функцію додавання двох рядків у стовпчик *add_column(S1, S2)*.

9.20. Скласти алгоритм множення „у стовпчик” двох чисел, записаних у вигляді рядків, що є позиційними записами цих чисел у десятковій системі числення. Використати підпрограми:

- 1) функцію *get_digit(c)* отримання цифри за символом c ;
- 2) функцію *get_symbol(d)* отримання символу за цифрою d ;
- 3) процедуру *mul_digit(n1, n2, p, n)* множення двох цифр $n1, n2$ з урахуванням перенесення p та отримання останньої цифри результату n ;

- 4) функцію *mul_str_char(s, c)* множення рядка s на символ c ;
- 5) функцію *add_string(s1, s2, n)* додавання двох рядків у стовпчик зі „зсувом” другого рядка на n позицій ліворуч.

9.21. Скласти процедуру "стискання" рядка: кожний підрядок, який складається з кількох входжень одного і того ж символу, замінюється самим цим символом.

9.22. Скласти підпрограми для

- а) підрахунку кількості слів рядка;
- б) отримання найдовшого слова;
- в) отримання найкоротшого слова;
- г) отримання всіх слів, які є паліндромами (симетричними);
- д) отримання всіх слів, які є ідентифікаторами;

д) отримання всіх слів, які є натуральними числами.

9.23. Відстанню Хемінга між двома рядками однакової довжини називається кількість позицій, на яких у цих двох рядках розташовано різні символи. Скласти функцію обчислення відстані Хемінга двох рядків.

Передбачити

а) нерекурсивний варіант

б) рекурсивний варіант

9.24. Відстань Левентштейна між двома рядками s_1 , s_2 - $Lev(s_1, s_2)$ визначається наступним чином:

Якщо один з рядків порожній, то відстань дорівнює довжині іншого рядка.

Якщо обидва рядки непорожні, то:

$$Lev(s1[:i], s2[:j]) = \begin{cases} Lev(s1[:i-1], s2[:j-1]), & \text{якщо } s1[i] = s2[j] \\ 1 + \min\{Lev(s1[:i-1], s2[:j-1]), Lev(s1[:i], s2[:j-1]), \\ Lev(s1[:i-1], s2[:j])\}, & \\ \text{якщо } s1[i] \neq s2[j] \end{cases}$$

при $i > 0$

Скласти функцію обчислення відстані Левенштейна двох рядків.

Передбачити

а) нерекурсивний варіант

б) рекурсивний варіант

Підказка (а): Використати матрицю цілих чисел з кількістю рядків, рівною довжині першого рядка, та кількістю стовпчиків, рівною довжині другого рядка. У елементах матриці з індексами (i, j) розмішувати відстані $Lev(s1[:i], s2[:j])$

9.25. Скласти рекурсивні підпрограми для обчислень значень функцій

$$а) \quad f(x) = \begin{cases} f\left(\frac{x}{2}\right) + \sqrt{x}, & \text{якщо } x \geq 10^{-6}, \\ 0, & \text{якщо } x < 10^{-6}. \end{cases}$$

$$\text{б) } f(x) = \begin{cases} f(x+1) + \frac{1}{x}, & \text{якщо } 10^{-6} \leq x \leq 10^3, \\ 0, & \text{якщо } x > 10^3, x < 10^{-6}. \end{cases}$$

$$\text{в) } f(x) = \begin{cases} f(\ln x) + \ln x, & \text{якщо } x > 10^{-6}, \\ 0, & \text{якщо } x \leq 10^{-6}. \end{cases}$$

9.26. Скласти рекурсивну функцію для обчислення многочленів Ерміта

$$H_0(x) = 1, H_1(x) = 2x,$$

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x), \quad n = 2, 3, \dots$$

Порівняти кількість дій у рекурсивному та нерекурсивному варіантах.

9.27. Визначити рекурсивну функцію обчислення *НСД*(n, m) натуральних чисел, яка ґрунтується на співвідношенні $НСД(n, m) = НСД(m, r)$, де r - остача від ділення n на m .

9.28. Визначити рекурсивну процедуру представлення натурального числа Z у вісімковій системі числення.

9.29. Визначити рекурсивну функцію обчислення степеня дійсного числа з цілим показником x^n згідно з формулою

$$x^n = \begin{cases} 1, & \text{якщо } n=0, \\ \frac{1}{x^{|n|}}, & \text{якщо } n < 0, \\ x * x^{n-1}, & \text{якщо } n > 0. \end{cases}$$

9.30. Визначити рекурсивну функцію для обчислення біноміального коефіцієнту $C_m^b, 0 \leq m \leq n$, за такою формулою:

$$C_n^0 = C_n^n = 1; C_n^m = C_{n-1}^m + C_{n-1}^{m-1}, \quad \text{при } 0 < m < n.$$

9.31. Визначити рекурсивну функцію для знаходження суми додатніх дійсних чисел, які складають непорожню послідовність, за якою слідує від'ємне число.

9.32. Числами Фібоначчі називається числова послідовність $\{F_n\}$, задана рекурентним співвідношенням другого порядку $F_0 = 0, F_1 = 1, F_k = F_{k-1} + F_{k-2}, k = 2, 3, \dots$ Визначити рекурсивну

функцію для обчислення числа Фібоначчі F_n для заданого натурального n . Порівняти працемісткість рекурсивного і нерекурсивного варіантів.

9.33. Задані натуральні числа a, c, m . Визначити рекурсивну функцію для обчислення $f(m)$ за формулою

$$f(m) = \begin{cases} m, & \text{якщо } 0 \leq m \leq 9, \\ g(m) * f(m-1-g(m)) + m, & \text{у інших випадках.} \end{cases}$$

$g(m)$ - остача від ділення $a * m + c$ на 10.

9.34. Визначити рекурсивні функції

- перевірки заданого рядка на симетричність;
- побудови рядка, інвертованого по відношенню до заданого;
- заміни у вихідному рядку всіх входжень даного символу даним рядком;
- перевірки, чи є один рядок початком іншого;
- перевірки на входження одного рядка у інший.

Вказівка. Нехай $\Lambda, A, B \in W$ (Λ - порожній рядок), $x, y \in Ch$.

Для побудови рекурсивних функцій використати співвідношення

а) $сим(\Lambda) = Icm$, $сим(add(x, \Lambda)) = Icm$,

$сим(app(add(x, A), y)) = (x=y) \& сим(A)$;

б) $інв(\Lambda) = \Lambda$,

$інв(add(x, A)) = app(інв(A), x)$;

в) $зам(\Lambda, x, B) = \Lambda$,

$зам(add(y, A), x, B) = add(y, зам(A, x, B))$,

$зам(add(x, A), x, B) = B + зам(A, x, B)$;

г) $поч(\Lambda, B) = Icm$, $поч(add(x, A), \Lambda) = Хиб$,

$поч(add(x, A), add(y, B)) = (x = y) \& поч(A, B)$;

д) $вход(\Lambda, B) = Icm$, $вход(add(x, A), \Lambda) = Хиб$,

$вход(add(x, A), add(y, B)) = поч(add(x, A), add(y, B)) \vee вход(add(x, A), B)$.

9.35. Скласти рекурсивну функцію для обчислення функції Аккермана $Акк(n, m)$, заданої співвідношенням

$Акк(0, m) = m + 1$;

$Акк(n, 0) = Акк(n - 1, 1)$;

$Акк(n, m) = Акк(n - 1, Акк(n, m - 1))$.

Обчислити $Акк(0, 5), Акк(1, 2), Акк(2, 2)$.

Покажемо спосіб обчислення функції Аккермана на прикладі:

$$Akk(1,2)=Akk(0,Akk(1,1))=Akk(0,Akk(0,Akk(1,0)))= \\ Akk(0,Akk(0,Akk(0,1)))=Akk(0,Akk(0,2))=Akk(0,3)=4.$$

9.36. Скласти рекурсивну функцію обчислення суми:

$$S_{ij} = \sum_{k_1=j-1}^{i-1} \sum_{k_2=j-2}^{k_1-1} \dots \sum_{k_{j-1}=1}^{k_{j-2}-1} \sum_{k_j=0}^{k_{j-1}-1} 1$$

9.37. Ханойські вежі. Дошка має три стрижні. На перший нанизано N дисків спадного догори діаметра. Потрібно, перекладаючи диски по одному, розмістити їх в початковому порядку на другому стрижні. При цьому більший диск ніколи не повинен розміщуватись над меншим.

Скласти підпрограму, яка ілюструє порядок переміщення дисків. Викликати її при $N=3$. Підрахувати кількість ходів, які потрібні для переміщення дисків. Знайти її залежність від N .

9.38. Скласти програму, яка відображає всі перестановки цілих чисел від 1 до N .

Вказівка. Множину перестановок цілих чисел від 1 до N можна отримати з множини всіх перестановок цілих чисел від 1 до $N-1$, вставляючи N в усі можливі позиції в кожній перестановці.

9.39. Скласти підпрограми зі змінною кількістю параметрів для обчислення функцій

$$\begin{aligned} \text{а) } f(x_1, \dots, x_n) &= \begin{cases} 0, & \text{якщо } x_1 > x_2 > \dots > x_n, \\ \sum_{i=1}^n |x_{i+1} - x_i|, & \text{в інших випадках;} \end{cases} \\ \text{б) } f(x_1, \dots, x_n) &= \begin{cases} 1, & \text{якщо } x_1 \leq x_2 \leq \dots \leq x_n, \\ \sum_{i=1}^{n-1} 2^{x_i + x_{i+1}}, & \text{в інших випадках;} \end{cases} \\ \text{в) } f(x_1, \dots, x_n) &= \begin{cases} 0, & \text{якщо } x_1 \leq 2^1 \leq x_2 \leq 2^2 \leq \dots \leq x_n \leq 2^n, \\ \prod_{i=1}^n x_i, & \text{в інших випадках;} \end{cases} \end{aligned}$$

$$\begin{aligned}
 \text{г) } f(x_1, \dots, x_n) &= \begin{cases} 1, & \text{якщо } \max_i x_i > \sum_{i=1}^n x_i, \\ \sum_{x_i > 0} x_i, & \text{в інших випадках;} \end{cases} \\
 \text{г) } f(x_1, \dots, x_n) &= \begin{cases} 0, & \text{якщо } \min_i x_i < \prod_{i=1}^n x_i, \\ \sum_{i=1}^{n-1} x_i * x_{i+1}, & \text{в інших випадках;} \end{cases} \\
 \text{д) } f(x_1, \dots, x_n, y_1, \dots, y_n) &= \sum_{i=1}^n (x_i^2 + y_i^2 + x_i * y_i);
 \end{aligned}$$

Вказівка: оформити x_i як позиційні, а y_i , - як ключові параметри.

$$\text{е) } f(x_1, \dots, x_n, y_1, \dots, y_n) = (x_1^2 + \dots + x_n^2) * (y_1^2 + \dots + y_n^2);$$

Вказівка: оформити x_i як позиційні, а y_i , - як ключові параметри.

$$\text{е) } f(x_1, \dots, x_n, y_1, \dots, y_n) = \prod_{i=1}^n (x_i^3 + y_i^3);$$

Вказівка: оформити x_i як позиційні, а y_i , - як ключові параметри.

$$\text{ж) } f(x_1, \dots, x_n, y_1, \dots, y_n) = \left(x_1 + \frac{1}{y_1}\right) * \dots * \left(x_n + \frac{1}{y_n}\right);$$

Вказівка: оформити x_i як позиційні, а y_i , - як ключові параметри.

$$\text{з) } f(x_0, x_1, \dots, x_n, y_1, \dots, y_n) = x_0 + \sum_{j=1}^n \left(x_j + \prod_{i=1}^j y_i\right);$$

Вказівка: оформити x_i як позиційні, а y_i , - як ключові параметри.

$$\begin{aligned}
 \text{і) } f(x_0, x_1, \dots, x_n, y_1, \dots, y_n) &= x_0 + x_1^2 * y_1^2 + \dots \\
 &\dots + x_n^2 * y_1^2 * y_2^2 * \dots * y_n^2
 \end{aligned}$$

Вказівка: оформити x_i як позиційні, а y_i , - як ключові параметри.

9.40. Визначити функцію для обчислення кореня рівняння $f(x)=0$ на відрізку $[a, b]$, на якому $f(x)$ змінює знак, з заданою точністю ε методом ділення відрізка навпіл. Виконати обчислення кореня для функції $f(x)=x^3-7*x-1$.

9.41. Визначити функцію для обчислення кореня рівняння $f(x)=0$ на відрізку $[a, b]$, на якому $f(x)$ змінює знак, з заданою точністю ε методом хорд. Виконати обчислення кореня для функції $f(x)=x^3-7*x-1$.

Вказівка. Обчислюючи корінь методом хорд, з'єднують прямою точки $(a, f(a))$ та $(b, f(b))$ та знаходять точку x перетину цієї прямої з віссю абсцис. Якщо знаки $f(a)$ та $f(x)$ співпадають, далі пошук проводять на відрізку $[x, b]$, інакше – на відрізку $[a, x]$.

9.42. Скласти підпрограми для обчислення визначеного інтегралу

а) методом прямокутників; б) методом Сімпсона.

Вказівка б). Для обчислення інтегралу використати границю

$$I = \int_a^b f(x) dx = \lim_{n \rightarrow \infty} S_n, \quad S_k = \frac{h_k}{3} (f_0 + 4f_1 + 2f_2 + 4f_3 + \dots \\ \dots + 4f_{n-3} + 2f_{n-2} + 4f_{n-1} + f_n),$$

де $f_i = f(a + i * h_k)$, $h_k = \frac{b-a}{n}$, $n = 2^k, i = 0, 1, \dots, n$, i

представлення S_k у вигляді $S_k = S_k^{(1)} + S_k^{(2)} + S_k^{(4)}$,

де

$$S_k^{(1)} = \frac{h_k}{3} (f_0 + f_n), \quad S_k^{(2)} = \frac{h_k}{3} (2f_2 + 2f_4 + \dots + 2f_{n-2}),$$

$$S_k^{(4)} = \frac{h_k}{3} (4f_1 + 4f_3 + \dots + 4f_{n-1}).$$

Для обчислення $S_k^{(1)}, S_k^{(2)}, S_k^{(4)}$ використати рекурентне співвідношення

$$S_k^{(1)} = \frac{1}{2} * S_{k-1}^{(1)}, \quad S_1^{(1)} = \frac{h}{3} (f(a) + f(b)), \quad h = \frac{b-a}{2}$$

$$S_k^{(2)} = \frac{1}{2} * S_{k-1}^{(2)} + \frac{1}{4} * S_{k-1}^{(4)}, S_1^{(2)} = 0, k = 2, 3, \dots$$

$$S_k^{(4)} = \frac{4h_k}{3} (f(a+h_k) + f(a+3h_k) + \dots + f(a+(n-1)h_k)),$$

$$S_1^{(4)} = \frac{4h}{3} f\left(\frac{a+b}{2}\right).$$

9.43. Нехай M_1, M_2, \dots, M_n - матеріальні точки, положення яких на площині задано координатами $(x_1, y_1), \dots, (x_n, y_n)$, а маси визначаються за допомогою вагової функції $g(x, y)$. Положення центру ваги цих точок задано формулами:

$$x = \frac{\sum_{i=1}^n g(x_i, y_i) x_i}{\sum_{i=1}^n g(x_i, y_i)}; \quad y = \frac{\sum_{i=1}^n g(x_i, y_i) y_i}{\sum_{i=1}^n g(x_i, y_i)}.$$

Визначити функцію обчислення точки центру ваги (x, y) при заданій ваговій функції $g(x, y)$.

Складену функцію використати для знаходження положення центра ваги n точок при $g(x, y) = \sqrt{x^2 + y^2}$.

9.44. Скласти підпрограму для знаходження елемента дійсного вектора, який задовольняє умову, задану булівською функцією $Q(x)$. Виконати пошук, коли умовою $Q(x)$ є:

а) $x=a$; б) $x>a$; в) $a<=x<=b$;

де a, b – задані числа.

9.45. Скласти функцію для обчислення скалярного добутку двох векторів з використанням λ -функції для обчислення добутку двох чисел.

Вказівка: використати функцію `sum()`

9.46. Скласти функцію для обчислення добутку матриці розміром $m \times n$ на вектор розміром n з використанням λ -функції.

Вказівка: використати функцію обчислення скалярного добутку двох векторів та спискутворення.

9.47. Скласти функцію для обчислення добутку вектору розміром m на матрицю розміром $m \times n$ з використанням λ -функції.

Вказівка: використати функцію обчислення скалярного добутку двох векторів та спискоутворення.

9.48. Скласти функцію для обчислення добутку матриці розміром $m \times n$ на матрицю розміром $n \times k$ з використанням λ -функції.

Вказівка: використати функцію обчислення скалярного добутку двох векторів та спискоутворення.

9.49. Скласти функцію для обчислення суми двох матриць розміром $m \times n$ з використанням λ -функції.

Вказівка: використати спискоутворення.

9.50. Скласти функції обчислення норм дійсної матриці порядку n з використанням λ -функції.

$$\text{а) } \|A\| = \max_{i=1, \dots, n} \sum_{j=1}^n |a_{ij}|; \quad \text{б) } \|A\| = \max_{j=1, \dots, n} \sum_{i=1}^n |a_{ij}|;$$

9.51. Скласти функцію, що перевіряє чи є задана цілочисельна квадратна матриця магічним квадратом, тобто такою, в якій суми елементів в усіх рядках і стовпчиках однакові. Використати λ -функцію.

Вказівка: використати функцію `zip()` для отримання транспонованої матриці.

9.52. Гральна карта K має два поля: *масть*, *значення* і представляється кортежем $K = (m, z)$, де $m \in \tilde{I}$, $z \in Z$. Відповідні кортежі означені наступним чином:

$$M = ("піки", "хрести", "бубни", "чирви"); \\ Z = ("шість", "сім", "вісім", "дев'ять", \\ "десять", "валет", "дама", "король", "туз").$$

Скласти програму, яка перевіряє, чи "б'є" карта K_1 карту K_2 , враховуючи те, що масть KM є козирною.

Передбачити :

- 1) опис функції $indL(name, L)$ – отримання елемента списку/кортежа L по введеному користувачем з клавіатури індексу. Параметр $name$ – пояснювальний текст. Призначення цієї функції – спростити і забезпечити захищене введення елемента з наперед

означеного кортежа|списка. Наприклад, введення окремої карти буде означенням кортежа $(indL("Мать", M), indL("Значення", Z))$.

- 2) опис функції $in_karta(nameK)$ – введення карти з ім'ям $nameK$ – рядок тексту, наприклад, $K1 = in_karta("K1")$.
- 3) опис функції $k_karta(k1, k2, km)$ – чи б'є карта $k1$ карту $k2$ при козирі km .

9.53. Скласти програму, яка знаходить з точністю $0 < \varepsilon < 1$ наближений розв'язок системи лінійних алгебраїчних рівнянь (СЛАР) $A\vec{x} = \vec{f}$ за допомогою ітераційного методу мінімальних нев'язок (ММН).

Алгоритм ММН: Задаємо необхідну точність ε , максимальну кількість ітерацій mki , довільний вектор $\vec{x}^{(0)}$, наприклад, $\vec{x}^{(0)} = \vec{f}$. Послідовно, для $k=0, 1, \dots, mki$ переходимо від $\vec{x}^{(k)}$ до наступного елемента наближення $\vec{x}^{(k+1)}$, використовуючи ланцюжок формул

$$\vec{r} = A * \vec{x}^{(k)} - \vec{f}, \quad \vec{y} = A * \vec{r}, \quad \tau = \frac{(\vec{y}, \vec{r})}{(\vec{y}, \vec{y})}, \quad \vec{x}^{(k+1)} = \vec{x}^{(k)} - \tau * \vec{r}.$$

Закінчення цього процесу – виконання умови

$$\left(\|\vec{x}^{(k+1)} - \vec{x}^{(k)}\|_c < \varepsilon \right) \vee err, \text{ де } err = (k \geq mki).$$

Для застосування ММН матриця СЛАР має бути симетрична ($A = A'$) і з діагональною перевагою ($|a_{ii}| > \sum_{j=u \neq i}^n |a_{ij}|, i = \overline{1, n}$).

Передбачити:

- 1) опис функції $mmn(A, f, x, eps=1e-8, mki=1001)$ – алгоритм ММН отримання x, err , де $x = \vec{x}^{(k+1)}$; $err = -1/0/1$ – матриця не задовольняє умові застосування ММН | знайдено розв'язок з необхідною точністю | за задану кількість ітерацій розв'язок з необхідною точністю не знайдено. Функція mmn повинна мати дві внутрішні функції:
 - перевірка матриці A на симетричність;
 - перевірка матриці A на діагональну перевагу.
- 2) опис функції get_vec – введення вектора з n елементів

- типу *int* | *float*.
- 3) опис функції *put_vec* – форматове виведення вектора (по 5 елементів в рядок для типу *float* | по 10 елементів в рядок для типу *int*).
 - 4) опис функції *dot(x,y)* – скалярного добутку двох векторів.
 - 5) опис функції множення матриці на вектор (з використанням функції *dot(x,y)*).
 - 6) опис функції знаходження лінійної комбінації двох векторів $p\vec{X} + q\vec{Y}$, де p, q – дійсні числа.
 - 7) опис функції знаходження норми вектора $\|\vec{x}\|_c = \max_i (|x_i|)$.
 - 8) опис функції *get_matr* – введення матриці розмірності $m \times n$ (з використанням функції *get_vec*).
 - 9) опис функції *put_matr* – форматове виведення матриці (з використанням функції *put_vec*).

9.54. Задано деяку послідовність натуральних чисел $N = [N_i]$, $N_i > 0$. Скажемо, що елемент в послідовності є *мажорантним*, якщо він зустрічається більше $[n/2]$ раз, де n – кількість елементів N . Написати функцію знаходження *Nmaj* – мажорантного елемента. Якщо такого не існує, то *Nmaj* = *None*. Написати функцію *get_lst(type_e, cond)* – введення з клавіатури послідовності N з елементами типу *type_e*, для яких виконується умова *cond(m)* (тобто, введене число m в послідовність попадає тільки при виконанні умови *cond* !). При звертанні до *get_lst* функцію *cond* оформити як анонімну. Написати *task1* – процедуру розв'язання поставленої задачі і виконати її тестування.

9.55. За певний період роботи інтернет-магазину задано інформацію про постачання товарів у вигляді послідовності рядків :

Ім'я_постачальника Товар Кількість Вартість

Написати процедуру *get_info()* – введення з клавіатури послідовності рядків з цією інформацією (пустий рядок = кінець вводу, Вартість в коп, Ім'я_постачальника, Товар – латиниця). По

- а) n -го члена арифметичної прогресії;
- б) суми n членів арифметичної прогресії.

9.60. Дано перший член і знаменник геометричної прогресії.

Визначити рекурсивні функції для знаходження:

- а) n -го члена геометричної прогресії;
- б) суми n членів геометричної прогресії.

9.61. Використовуючи формулу

$$\text{НСК}(m, n) = \frac{m \cdot n}{\text{НСД}(m, n)}$$

де $\text{НСД}(m, n)$ – найбільших спільний дільник чисел m і n , скласти програму для обчислення найменшого спільного кратного двох натуральних чисел m і n .

9.62. Перевірити, чи існують досконалі числа з проміжку $[2, n]$, котрі є повними квадратами. Якщо так, то вивести на екран перше з них. Число називається досконалим, якщо воно дорівнює сумі всіх своїх дільників, крім самого цього числа (наприклад, числа 6 і 28: $6 = 1 + 2 + 3$, $28 = 1 + 2 + 4 + 7 + 14$).

9.63. Обчисліть $A + B$ у двійковій системі числення.

Вказівка: Опишіть функції для переведення числа з двійкової системи числення у десяткову та навпаки.

9.64. Число назвемо "дзеркально простим", якщо воно є простим, і простим є число, записане тими ж цифрами у зворотному порядку. Знайти кількість "дзеркально простих" чисел на проміжку від a до b .

Вказівка: Опишіть дві підпрограми, перша з яких інвертує задане число, інша – визначає чи є число простим.

9.65. Перерахуйте усі розбиття цілого додатного числа n на цілі додатні доданки. Розбиття повинні мати наступні властивості:

- Доданки у розбиттях йдуть у незростаючому порядку.
- Розбиття перераховуються у лексикографічному порядку.

Вказівка: Опишіть відповідну рекурсивну функцію.

Тема 10 Модулі

10.1. Описати модуль роботи з відрізками на числовій вісі. Тип відрізка представити у вигляді кортежу:

(a, b, empty)

де a, b - границі відрізка, empty - ознака того, що відрізок порожній.

Реалізувати дії над відрізками:

- 1) зробити відрізок t порожнім;
- 2) чи порожній відрізок t ;
- 3) покласти відрізок t рівним a, b ;
- 4) покласти відрізок t рівним перетину відрізків $t1, t2$.

З використанням модуля скласти програму розв'язку системи квадратних нерівностей вигляду $x^2 + p_i x + q_i < 0$. Пари коефіцієнтів нерівностей p_i, q_i вводяться з пристрою введення.

10.2. Описати модуль для реалізації мультимножини цілих чисел на базі словника. Мультимножина - це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість входжень.

Кількість входжень елемента k ($0 \leq k \leq n$) у мультимножину - це значення елемента словника з ключем k .

Реалізувати дії над мультимножинами:

- 1) зробити мультимножину порожньою;
- 2) чи є мультимножина порожньою;
- 3) додати елемент до мультимножини;
- 4) забрати елемент з мультимножини (кількість входжень елемента зменшується на 1, якщо елемент не входить - відмова);
- 5) кількість входжень елемента у мультимножину;
- 6) об'єднання двох мультимножин (в результаті об'єднання кількість входжень елемента визначається як максимальна з двох мультимножин);

7) перетин двох мультимножин (в результаті кількість входжень елемента визначається як мінімальна з двох мультимножин);

З використанням модуля розв'язати задачі:

а) знайти символ, який входить у рядок S максимальну кількість разів;

б) перевірити, чи складаються рядки $S1, S2$ з одних і тих же символів, які входять у ці рядки однаково кількість разів;

в) перевірити, чи вірно, що всі символи рядка $S1$, входять також у рядок $S2$, причому не меншу кількість разів, ніж у $S1$.

10.3. Описати модуль роботи з точками та відрізками на площині. Типи точки та відрізка представити у вигляді кортежів:

Точка – (x, y) ,

Відрізок – (a, b) ,

де a, b – точки

Реалізувати дії над точками:

1) взяти точку t ;

2) покласти точку t рівною (x, y) ;

3) показати точку t .

Реалізувати дії над відрізками:

1) взяти відрізок s ;

2) показати відрізок s ;

3) покласти відрізок s рівним a, b ;

4) довжина відрізка s ;

5) чи лежить точка t на одній прямій з відрізком s ;

6) чи лежить точка t всередині відрізка s ;

7) площа трикутника, утвореного точкою t та відрізком s .

Вводиться послідовність точок. З використанням модуля роботи з точками та відрізками на площині знайти:

а) трикутник з найбільшою площею, утворений точками послідовності;

б) коло найменшого радіуса, всередині якого лежать всі точки послідовності;

в) відрізок, на якому лежить найбільша кількість точок послідовності;

г) коло, на якому лежить найбільша кількість точок послідовності.

10.4. Описати модуль та скласти програму для реалізації гри у “хрестики-нолики” на полі розміром 3x3. У модулі реалізувати дії:

- 1) зробити хід гравця;
- 2) зробити хід комп'ютера;
- 3) показати ігрове поле.
- 4) перевірити, чи закінчена гра, та повернути переможця, якщо

є.

10.5. Відома гра у “морський бій” полягає у наступному. Два гравці на двох полях 10x10 розставляють „кораблі” – прямокутники (4 – розміром 1x1, 3 – розміром 2x1, 2 – розміром 3x1, 1 – розміром 4x1). Кораблі не можуть мати сусідніх клітин по горизонталі, вертикалі або діагоналі. Гравці не бачать розстановку кораблів супротивника. Потім гравці по черзі роблять ходи (кожний хід – це вказання клітини на полі). Якщо гравець потрапляє на поле, яке займає корабель супротивника, він має право на позачерговий хід. Виграє той, хто першим знешкодить всі кораблі супротивника.

Описати модуль та програму для реалізації гри у “морський бій” між гравцем та комп'ютером. Передбачити у модулі реалізацію дій:

- 1) додати корабель;
- 2) зробити хід гравця;
- 3) зробити хід комп'ютера;
- 4) показати ігрове поле (поле гравця та стан поля комп'ютера).

10.6. Описати модуль роботи з квадратними матрицями та векторами. Реалізувати дії:

- 1) Ввести матрицю
- 2) Вивести матрицю
- 3) Множення матриць
- 4) Множення матриці на вектор
- 5) Множення вектора на матрицю
- 6) Переставлення рядків матриці
- 7) Переставлення стовпчиків матриці
- 8) Отримання рядка матриці
- 9) Множення вектора на число

10) Віднімання вектора від всіх рядків матриці

З використанням модуля розв'язати задачі:

а) Перетворити матрицю у верхню трикутну лінійними перетвореннями

б) Визначити ранг матриці

в) Обчислити визначник матриці

г) Обчислити обернену матрицю

10.7. Описати модуль роботи з датами. Дата для обробки задається кортежем (<день>, <місяць>, <рік>).

Реалізувати функції:

- `input_date()` - введення дати з клавіатури. При введенні дата задається рядком `dd.mm.yyyy` Функція повертає дату у вигляді кортежу.
- `print_date(d)` – виведення дати. `d` – кортеж дати. Дата виводиться у вигляді рядка `dd.mm.yyyy`
- `is_less(d1, d2)` – чи менше дата `d1` ніж `d2`. `d1, d2` – кортежі дат. Повертає булівське значення.
- `nexth_day(d)` – дата наступного дня. `d` – кортеж дати. Повертає дату наступного дня (кортеж).

З використанням модуля розв'язати задачу: ввести 2 дати `d1`, `d2` та знайти кількість днів між `d1` та `d2` (з + або -).

10.8. Побудувати модуль який виконує дії над раціональними числами. Раціональне число представити кортежем з 2 елементів: цілий чисельник та натуральний знаменник.

Реалізувати функції:

- Додавання 2 раціональних чисел `add(a, b)`
- Множення 2 раціональних чисел `mul(a, b)`
- Віднімання 2 раціональних чисел `sub(a, b)`
- Ділення 2 раціональних чисел `div(a, b)`
- Створення раціонального числа з чисельника `m` та знаменника `n` `create(m, n)`
- Введення раціонального числа `inp()`
- Виведення раціонального числа `prt(a)`
- Перевірка відношення `a == b` – `eq(a, b)`

- Перевірка відношення $a < b - \text{lt}(a, b)$

Також реалізувати внутрішню функцію, яка виконує скорочення чисельника та знаменника на їх НСД `_reduce(a)`

Усі функції, окрім `prt`, `eq`, `lt` повертають раціональне число, яке є результатом відповідної операції.

Модуль також має містити глобальну змінну `error`, яка дорівнює 0, якщо остання дія виконана правильно та не дорівнює 0, якщо у останній дії була помилка. Приклад помилки – нульовий знаменник. На початку роботи з модулем значення цієї змінної – 0.

Модуль також має містити функцію `clear_error()`, яка встановлює значення змінної `error` у 0.

Якщо під час виконання будь-якої функції виникла помилка, ця функція має повернути кортеж `(None, None)`

З використанням цього модуля розв'язати задачу: ввести k раціональних чисел, порахувати їх суму s та обчислити значення $1/s$.

10.9. Реалізувати модуль для роботи з поліномами $P_n(x)$ степеня n . Поліном реалізувати як список коефіцієнтів. У модулі реалізувати такі операції:

- 1) зчитати поліном $P_n(x)$;
- 2) показати поліном $P_n(x)$;
- 3) визначити значення полінома $P_n(x)$ для заданого x ;
- 4) знайти похідну $P'_n(x)$ від полінома $P_n(x)$;
- 5) знайти первісну $F_n(x)$ полінома $P_n(x)$, що задовольняє умову $F_n(0) = c$, де c – задане дійсне число;
- 6) знайти суму двох поліномів;
- 7) знайти добуток двох поліномів;
- 8) реалізувати операцію ділення з остачею двох поліномів.

Використовуючи описаний модуль, розв'язати задачі:

- а) відстань, яку пройшла матеріальна точка за час t визначається за законом $s(t) = P_n(t)$; визначити її швидкість у момент часу t ;

- б) швидкість руху точки в момент часу t визначається за законом $v(t) = P_n(t)$; визначити яку відстань пройде точка за відрізок часу $[t_1, t_2]$;
- в) знайти похідну порядку k від полінома $P_n(x)$;
- г) визначити найбільший спільний дільник двох поліномів $P_n(x)$ і $R_n(x)$;
- д) визначити найменше спільне кратне двох поліномів;
- е) розв'язати задачу Коші для звичайного диференціального рівняння
- $$y^{(k)} = P_n(x), \quad x \geq x_0, \quad y(x_0) = y_0, \dots, y^{(k-1)}(x_0) = y_0^{k-1}.$$

Тема 11 Множини

.....

11.1. Скласти програму, що визначає мінімальний елемент числової множини.

11.2. Скласти програму виведення елементів числової множини в порядку зростання.

11.3. Визначити функцію доповнення множини, якщо відома універсальна множина.

11.4. Вагою числової множини називається сума модулів всіх його елементів. Вага порожньої множини вважається рівною нулеві. Скласти програму для обчислення ваги числової множини.

11.5. Діаметром числової множини називається величина

$$d(A) = \max_{x, y \in A} |x - y|.$$

Скласти програму обчислення діаметра $d(A)$.

11.6. Дано n цілих чисел від 1 до 50. Визначити, скільки серед них чисел Фібоначчі та скільки чисел, перша значуща цифра в десятковому запису яких 1 чи 2. Надрукувати всі ці числа (без повторів) у порядку зростання.

11.7. Скласти програми

а) обчислення кількості різних (значущих) цифр у десятковому записі натурального числа n ;

б) друку у порядку зростання всіх цифр, які не входять до десяткового запису натурального числа n .

11.8. Скласти програму визначення тривалості місяця у році Y .
Вказівка. Використати в якості базового типу в означенні множини

січ, лют, берез, квіт, трав, черв, лип, серп, верес, жовт, лист, груд;.

11.9. Дана непорожня послідовність символів (рядок) S . Описати функцію, яка визначає загальну кількість цифр та знаків операцій, що входять до рядка S .

11.10. Дано рядок з малих латинських літер. Надрукувати:

а) перші входження літер до рядка, зберігаючи початковий взаємний порядок;

б) всі літери, що входять до рядка не менше двох раз;

в) всі літери, що входять до рядка по одному разу.

11.11. Дано рядок символів S . В алфавітному порядку надрукувати (по одному разу) всі малі українські голосні літери, що входять до S .

11.12. Дана непорожня послідовність слів із малих українських літер; між сусідніми словами - пропуск, за останнім символом - крапка. Надрукувати в алфавітному порядку:

а) всі голосні літери, які входять до кожного слова;

б) всі приголосні літери, які не входять до жодного слова;

в) всі дзвінки приголосні літери, які не входять до жодного слова;

г) всі глухі приголосні, які не входять до жодного слова;

д) всі приголосні, які входять тільки в одне слово;

е) всі глухі приголосні літери, які не входять тільки в одне слово;

є) всі голосні літери, які входять не більше ніж в одне слово;

ж) всі дзвінки приголосні літери, які входять в кожне непарне слово і не входять в жодне парне слово;

з) всі глухі приголосні літери, які входять хоча б в одне непарне слово та не входять в жодне непарне слово.

Вказівка. До голосних літер належать а, е, и, і, о, у, є, ю, ї; до приголосних - всі інші крім й, ь; дзвінки приголосні - б, в, г, д, ж, з, л, м, н, р; глухі приголосні: к, п, с, т, ф, х, ц, ч, ш, щ.

11.13. Заданий вектор, компонентами якого є множини всіх продуктів із вказаного асортименту, що знаходяться у відповідному магазині. Визначити процедуру пошуку множини продуктів

- а) що є в кожному магазині;
- б) кожен з яких є хоча б в одному магазині;
- в) яких немає в жодному магазині.

11.14. Заданий вектор ГРУПА, компонентами якого $\text{ГРУПА}(x)$ є множини імен людей із вказаного списку тих, що були в гостях у людини з ім'ям X ($X \notin \text{ГРУПА}[X]$).

Визначити процедуру пошуку хоча б одної людини, яка була в гостях в усіх інших людей, імена яких містяться в компонентах вектора ГРУПА.

11.15. Дано n цілих чисел від 1 до m (можливо – з повторами). Визначити, скільки серед них чисел Фібоначчі. Надрукувати всі ці числа (без повторів) у порядку зростання. (використати множини).

11.16. Скласти функцію, яка за заданим рядком повертає список слів рядка без повторів. Використати множини. Вважати, що слова розділяються одним або декількома пропусками.

11.17. Кілька людей грає в гру «міста». Реалізуйте програму, що буде контролювати, чи назване місто не використовувалося раніше.

Тема 12 Файли

.....

12.1. Скласти програму для обчислення значень многочлена, його похідної, використовуючи файл його коефіцієнтів.

12.2. Дано файл, компоненти якого є дійсними числами. Скласти підпрограму для обчислення:

- а) суми компонент файлу;
- б) кількості від'ємних компонент файлу;
- г) останньої компоненти;
- д) найбільшого із значень компонент файлу;

- е) найменшого із значень компонент файлу з парними номерами;
- є) суми найбільшого і найменшого із значень компонент;
- ж) різниці першої і останньої компоненти файлу;
- з) кількості компонент файлу, які менші за середнє арифметичне усіх його компонент.

12.3. Дано файл, компоненти якого є цілими числами. Скласти підпрограму для обчислення:

- а) кількості парних чисел серед компонент;
- б) кількості квадратів непарних чисел серед компонент;
- в) різниці між найбільшим парним і найменшим непарним числами з компонент;
- г) кількості компонент в найдовшій зростаючій послідовності компонент файлу.

12.4. За масивом A , елементи якого утворені за законом

- а) $a_i=i$; б) $a_i=i^2$; в) $a_i=i!$; г) $a_i=2^{i+1}$; д) $a_i=2^i+3^{i+1}$ ($i=1,2,\dots,n$)
- побудувати файл, який містить елементи масиву A .

12.5. Послідовність x_1, x_2, \dots утворена за законом

$$x_i = \frac{i - 0.1}{i^3 + |\operatorname{tg} 2i|}, \quad i = 1, 2, \dots$$

Дано дійсне $\varepsilon > 0$. Побудувати файл, який містив би усі члени послідовності x_1, x_2, \dots , які стоять перед першим членом, для якого виконано: $|x_i| < \varepsilon$.

12.6. Скласти процедуру запису в файл всіх чисел Фібоначі, які не перевищують натуральне число n .

12.7. Дано файл f , компоненти якого є цілими числами. Побудувати файл g , який містив би всі компоненти файлу f :

- а) які є парними числами;
- б) які діляться на 3 і на 5;
- в) які є точними квадратами;
- г) записані у зворотному порядку;
- д) за виключенням повторних входжень одного і того самого числа.

12.8. Використовуючи файл f , компоненти якого є цілими числами, побудувати файл g , який містить всі парні числа файлу f і файл h - всі непарні. Послідовність чисел зберігається.

12.9. Задані натуральне число i файл f , компоненти якого є цілими числами. Побудувати файл g , записавши в нього найбільше значення перших n компонент файлу f , потім наступних n компонент і т.д. Розглянути два випадки:

- а) число компонент файлу ділиться на n ;
- б) число компонент файлу не ділиться на n .

В цьому випадку остання компонента файлу g повинна дорівнювати найбільшій із компонент файлу f , які утворюють останню (неповну) групу.

12.10. Дано файл f , компоненти якого є цілими числами. Файл f містить рівне число додатних та від'ємних чисел. Використовуючи допоміжний файл h , переписати компоненти файлу f в файл g так, щоб у файлі g :

- а) не було двох сусідніх чисел одного знаку;
- б) спочатку йшли додатні, потім від'ємні числа;
- в) числа йшли таким чином: два додатних, два від'ємних і т.д. (припускається, що число компонент в файлі f ділиться на 4).

12.11. По рядку S побудувати текстовий файл, який містить символи цього рядка.

12.12. Дано текстовий файл f . Побудувати файл g , утворений із файлу f :

- а) зміною всіх його великих літер однойменними малими;
- б) записом його компонент у зворотному порядку.

12.13. Дано текстовий файл, який складається не менш, ніж із 2 компонент. Визначити, чи є два перших символу файлу цифрами. Якщо так, то виявити, чи є число утворене цими цифрами парним.

12.14. Задано текстові файли $f1$ і $f2$. Переписати із збереженням послідовності компоненти файлу $f1$ у $f2$, а компоненти файлу $f2$ - у $f1$. Використати допоміжний файл h .

12.15. Задано текстові файли f і g . Записати в файл h спочатку компоненти файлу f , потім - компоненти файлу g зі збереженням порядку.

12.16. Дано текстовий файл. Скласти підпрограму для:

- а) додавання в його кінець заданого символу;
- б) додавання в його початок заданого символу;
- в) підрахунку числа входжень у файл заданого символу;

- г) визначення входження у файл заданої комбінації символів;
- д) вилучення заданого символу;
- е) вилучення других входжень кожного символу.

12.17. Скласти функцію перевірки рівності файлів, виконаної за один перегляд їх змісту.

12.18. Скласти функцію побудови множини символів текстового файлу.

12.19. Скласти процедуру злиття двох впорядкованих файлів у новий впорядкований файл.

12.20. Багаж пасажирів характеризується номером пасажирів, кількістю речей та їх загальною вагою. Дано файл пасажирів, який містить прізвища пасажирів, та файл, який містить інформацію про багаж декількох пасажирів (номер пасажирів – це номер запису у файлі пасажирів). Скласти процедуру для

а) знаходження пасажирів, у багажу якого середня вага однієї речі відрізняється не більш як на 1 кг від загальної середньої ваги речей;

б) визначення пасажирів, які мають більше двох речей, і пасажирів, кількість речей яких перевищує середнє число речей;

в) видачі відомостей про пасажирів, число речей у багажі якого не менш, ніж у будь-якому іншому багажі, а вага речей не більша, ніж у будь-якому іншому багажі з цим же числом речей;

г) визначення, чи мають хоча б два пасажирів багажі, які не відрізняються кількістю речей та відрізняються вагою не більше, ніж на 1 кг (якщо такі пасажирів є, то показати їх прізвища);

д) визначення пасажирів, багаж якого складається з однієї речі, вагою не менше 30 кг.

12.21. Дано файл, який містить різні дати. Кожна дата - це число, місяць та рік. Скласти процедури пошуку

- а) року з найменшим номером;
- б) всіх весняних дат;
- в) найпізнішої дати.

12.22. Дано файл, який містить відомості про книги. Відомості про кожну книгу - це прізвище автора, назва та рік видання. Скласти процедури пошуку

- а) назв книг даного автора, виданих з 1960 р.;

б) книг із заданою назвою. Якщо така книга є, то надрукувати прізвища авторів та рік видання.

12.23. Дано файл, який містить номери телефонів співробітників установи: вказуються прізвище співробітника, його ініціали та номер телефона. Визначити процедуру пошуку телефона співробітника за його прізвищем та ініціалами.

12.24. Дано файл, який містить відомості про кубики: розмір кожного кубуку (довжина ребра в см), його колір (червоний, жовтий, зелений, синій) та матеріал (дерев'яний, металевий, картонний). Скласти процедури пошуку

а) кількості кубиків кожного з перелічених кольорів та їх сумарний об'єм;

б) кількості дерев'яних кубиків із ребром 3 см та кількості металевих кубиків з ребром більшим за 5 см.

12.25. Дано файл, який містить відомості про речовину: вказується назва речовини, її питома вага та провідність (провідник, напівпровідник, ізолятор). Скласти процедури для

а) знаходження питомої ваги та назви всіх провідників;

б) вибору даних про напівпровідники та впорядкування їх, щоб кожний наступний був з меншою питомою вагою, ніж попередній.

12.26. Дано файл, який містить відомості про товари, що експортуються. Вказано назву товару, країну, яка імпортує товар, та об'єм прибулої частини у штуках. Скласти процедуру пошуку країн, в які екпортується даний товар та загальний об'єм його експорту.

12.27. Задано два файли $f1$ і $f2$. Файл $f1$ - це інвентарний файл, який містить відомості про те, скільки виробів і яких видів продукції зберігається на складі (вид продукції задається його порядковим номером). Файл $f2$ - це допоміжний файл, який містить відомості про те, наскільки зменшилась чи збільшилась кількість виробів по деяких видах продукції. Допоміжний файл може містити декілька відомостей про продукції одного виду або не містити жодної такої відомості. Скласти процедуру оновлення інвентарного файлу за допоміжним, склавши новий файл g .

12.28. Дано файл, який містить відомості про іграшки: вказується назва іграшки (наприклад: м'яч, лялька, конструктор і

т.д.), її вартість в гривнях та вікові границі дітей, для яких іграшка призначається (наприклад, для дітей від двох до п'яти років). Скласти процедури

а) пошуку назв іграшок, вартість яких не перевищує 40 грн. та які підходять дітям 5-и років;

б) пошуку назв іграшок, які підходять дітям і 4, і 10 років;

в) пошуку назв найдорожчих іграшок (ціна яких відрізняється від ціни найдорожчої іграшки не більше, ніж на 50 грн.);

г) визначення ціни найдорожчого конструктора;

д) визначення ціни всіх кубиків;

е) пошуку двох іграшок, які підходять дитині 3-х років, сумарна вартість яких не перевищує 20 грн.

є) пошуку конструктора ціною 22 грн., призначеного дітям від 5 до 10 років. Якщо такої іграшки немає, то занести відомості про неї у файл.

12.29. Дано файл, який містить відомості про прямокутники: вказується номер прямокутника у файлі, координати верхнього лівого кута прямокутника, координати нижнього правого кута прямокутника. Скласти процедуру пошуку прямокутника з найбільшою площею і визначення цієї площі.

12.30. Дано текстовий файл. Групи символів, які відокремлені пропусками (одним або декількома) і не містять пропусків усередині себе, будемо називати словами. Скласти підпрограми для

а) знаходження найдовшого слова у файлі;

б) визначення кількості слів у файлі;

в) вилучення з файлу усіх слів, які складаються з однієї букви, та зайвих пропусків;

г) вилучення всіх пропусків на початку рядків, в кінці рядків та між словами (окрім одного);

д) вставки пропусків у рядки рівномірно між словами так, щоб довжина всіх рядків (якщо в них більше 1 слова) була 80 символів та кількість пропусків між словами у одному рядку відрізнялась не більше, ніж на 1 (вважати, що рядки файлу мають не більше, ніж 80 символів).

Результат записати у файл *h*.

12.31. Визначити процедуру, яка утворює текстовий файл із 9 рядків, в першому з яких одна літера "1", в другому - дві літери "2", ... , в дев'ятому – дев'ять літер "9".

12.32. Визначити процедуру, яка за заданою послідовністю символів формує текстовий файл із рядками по 40 літер (в останньому рядку літер може бути і менше).

12.33. Визначити функцію:

- а) яка підраховує кількість порожніх рядків;
- б) яка обчислює максимальну довжину рядків текстового файлу.

12.34. Визначити процедуру виведення

- а) всіх рядків текстового файлу;
- б) рядків, які містять більше 60 символів.

12.35. Визначити функцію, що обчислює кількість рядків текстового файлу, які

- а) починаються із заданого символу;
- б) закінчуються на заданий символ;
- в) починаються і закінчуються одним і тим же символом;
- г) складаються з однакових символів.

12.36. Визначити процедуру, яка переписує у текстовий файл g всі рядки текстового файлу f

- а) із заміною в них символу '0' на '1' і навпаки;
- б) у інвертованому вигляді.

12.37. Визначити процедуру пошуку найдовшого рядка у текстовому файлі. Якщо таких рядків декілька, знайти перший з них.

12.38. Визначити процедуру, яка переписує компоненти текстового файлу f у файл g , вставляючи до початку кожного рядка один символ пропуску. Порядок компонент не повинен змінюватися.

12.39. Визначити процедуру, яка друкує по рядках зміст текстового файлу, вставляючи до початку кожного рядка його порядковий номер і символ пропуску.

12.40. Визначити процедуру пошуку у текстовому файлі рядків, фрагментом яких є заданий рядок.

12.41. Вважаючи, що довжина рядка текстового файлу f не перевищує 80, визначити процедуру, яка, доповнюючи короткі

рядки файлу f пропусками справа, формує текстовий файл g , всі рядки якого мають довжину 80.

12.42. Визначити процедуру, яка, виключаючи пропуски, що стоять на початку рядків текстового файлу f , формує текстовий файл g .

12.43. У текстовому файлі записана непорожня послідовність дійсних чисел, які розділяються пропусками. Визначити функцію обчислення найбільшого з цих чисел.

12.44. У текстовому файлі f записана послідовність цілих чисел, які розділяються пропусками. Визначити процедуру запису у текстовий файл g всіх додатних чисел із f .

12.45. У текстовому файлі кожний рядок містить декілька натуральних чисел, які розділяються пропусками. Числа визначають вигляд геометричної фігури (номер) та її розміри. Прийняті наступні домовленості:

- відрізок прямої задається координатами своїх кінців та має номер 1;
- прямокутник задається координатами верхнього лівого і нижнього правого кута та має номер 2;
- коло задається координатами центра і радіусом та має номер 3.

Визначити процедуру обчислення

а) відрізка з найбільшою довжиною;

б) прямокутника з найбільшим периметром;

в) кола з найменшою площею.

12.46. Скласти програму для перевірки правильності розстановки круглих, квадратних та фігурних дужок у текстовому файлі – програмі у Python. Вважати, що коментарів у програмі немає.

12.47. Скласти програму для перевірки правильності розстановки відступів у текстовому файлі – програмі у Python. Відступ визначається кількістю пропусків у рядку перед першим символом, що не є пропуском. Вважати, що коментарів у програмі немає та жоден логічний рядок не переходить на наступний фізичний рядок.

Підказка: Відступ повинен збільшуватись по відношенню до поточного після двокрапки ':' у кінці рядка. Після будь-якого

рядка, що не містить у кінці двокрапку, відступ (якщо він більше нуля) може зменшуватись. Тоді більші відступи «закривають». Після зменшення відступ повинен дорівнювати одному з відступів, які зустрічались та не є закритими.

Тема 13 Класи та об'єкти

13.1. Клас Person описано наступним чином:

```
class Person:                                #Клас Особа
    def __init__(self):
        self.name = None                    #прізвище
        self.byear = None                  #рік народження

    def input(self):                         #ввести особу
        self.name = input('Прізвище: ')
        self.byear = input('Рік народження: ')

    def print(self):                         #вивести особу
        print(self.name, self.byear, end = ' ')
```

Описати клас Знайомий на базі класу Person.

У цьому класі повинно бути як мінімум одне додаткове поле «номер телефону» а також методи введення та виведення інформації про знайомого.

Використати цей клас для побудови телефонного довідника.

Передбачити дії: створення довідника, додавання запису про знайомого, пошуку номера телефону за прізвищем та заміни номера телефону.

Телефонний довідник зберігати у файлі та читати з файлу.

Вказівка. Телефонний довідник представити у вигляді списку знайомих.

13.2. Описати клас Співробітник на базі класу Person (див. завдання 13.1).

У цьому класі повинно бути як мінімум два додаткових поля: «табельний номер» та «заробітна платня», а також методи введення та виведення інформації про співробітника та розрахунку заробітної платні за кількістю відпрацьованих годин та загальною кількістю

годин на місяць.

Використати цей клас для розрахунку заробітної платні за місяць, якщо співробітник працює на умовах погодинної оплати. Для розрахунку вводити відпрацьовані години по днях місяця (табелі). Вважається, що відома кількість годин, яку повинен відпрацювати співробітник кожного дня, для отримання 100% платні.

Передбачити дії: введення таблицю для співробітника, введення таблиців всіх співробітників, розрахунку заробітної платні всіх співробітників.

Табель зберігати у файлі та читати з файлу.

Вказівка. Табелі представити у вигляді словника співробітників з ключем «табеліний номер».

13.3. Описати клас Гість на базі класу Person (див. завдання 13.1) та клас ТипНомера з полями «назва типу», «плата за добу».

У класі Гість повинно бути як мінімум два додаткових поля: «номер» та «кількість днів проживання», а також методи введення та виведення інформації про гостя та розрахунку плати за проживання.

Використати цей клас для реєстрації гостя у номері та розрахунку плати за проживання усіх гостей.

Номери представити у вигляді списку кортежів (номер, тип номера).

Номери та інформацію про гостей зберігати у файлах та читати з файлів.

Вказівка. Інформацію про гостей представити у вигляді списку.

13.4. Описати клас Пасажир на базі класу Person (див. завдання 13.1).

У класі Пасажир повинно бути як мінімум два додаткових поля: «місто відправлення» та «місто прибуття», а також методи введення та виведення інформації про пасажирів та розрахунку плати за білет.

Використати цей клас для розрахунку плати за білети усіх пасажирів. Вважати що маршрути зберігаються у списку кортежів (місто1, місто2, відстань), а також те, що плата за білет пропорційна відстані та відома плата за 1 км відстані.

Вказівка. Інформацію про пасажирів представити у вигляді списку.

13.5. Описати клас Водій на базі класу Person (див. завдання

13.1).

У класі Водій повинно бути як мінімум два додаткових поля: «плата за тоно-кілометр» та «вантажопідйомність автомобіля», а також методи введення та виведення інформації про водія та розрахунку плати за перевезення вантажу на задану відстань.

Маршрутний лист водія за певну дату містить кількість кілометрів перевезень вантажу.

Використати цей клас для розрахунку плати всіх водіїв за період. Вважати що маршрутні листи водіїв зберігаються у списку кортежів (водій, дата, відстань), а також те, що плата за перевезення пропорційна добутку відстані на вагу вантажу.

13.6. Описати клас Поліном та реалізувати методи: введення поліному, виведення поліному, обчислення значення поліному у точці x , взяття похідної поліному, суми, різниці та добутку поліномів.

Використати цей клас для розв'язання задачі: ввести 2 поліноми P_1 , P_2 та рядок, який містить вираз, що залежить від 2 поліномів. Наприклад,

$$P_1 + P_2 * P_1 - P_2$$

Обчислити поліном, який буде значенням цього виразу.

Вказівка. Поліном представити у вигляді словника.

13.7. Описати клас для роботи з відрізками на числовій осі. Для відрізка передбачити поля:

(a, b, empty),

де a, b - границі відрізка, empty - ознака того, що відрізок порожній.

Реалізувати методи:

- 1) зробити відрізок t порожнім;
- 2) чи порожній відрізок t;
- 3) покласти відрізок t рівним a, b;
- 4) покласти відрізок t рівним перетину відрізків t1, t2.

З використанням класу скласти програму розв'язку системи квадратних нерівностей вигляду $x^2 + p_i x + q_i < 0$. Пари коефіцієнтів нерівностей (p_i, q_i) вводяться з пристрою введення.

13.8. Описати клас для реалізації мультимножини цілих чисел на базі словника. Мультимножина - це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість

входжень.

Кількість входжень елемента k ($0 \leq k \leq n$) у мультимножину - це значення елемента словника з ключем k .

Реалізувати методи:

- 1) зробити мультимножину порожньою;
- 2) чи є мультимножина порожньою;
- 3) додати елемент до мультимножини;
- 4) забрати елемент з мультимножини (кількість входжень елемента зменшується на 1, якщо елемент не входить - відмова);
- 5) кількість входжень елемента у мультимножину;
- 6) об'єднання двох мультимножин (в результаті об'єднання кількість входжень елемента визначається як максимальна з двох мультимножин);
- 7) перетин двох мультимножин (в результаті кількість входжень елемента визначається як мінімальна з двох мультимножин);

З використанням класу розв'язати задачі:

- а) знайти символ, який входить у рядок S максимальну кількість разів;
- б) перевірити, чи складаються рядки $S1$, $S2$ з одних і тих же символів, які входять у ці рядки однаково кількість разів;
- в) перевірити, чи вірно, що всі символи рядка $S1$, входять також у рядок $S2$, причому не меншу кількість разів, ніж у $S1$.

13.9. Описати класи для роботи з точками та відрізками на площині. Для точки та відрізка передбачити такі поля:

Точка – (x, y) , де x, y – дійсні числа;

Відрізок – (a, b) , де a, b – точки.

Реалізувати методи:

- 1) взяти точку t ;
- 2) покласти точку t рівною (x, y) ;
- 3) показати точку t .

Реалізувати методи:

- 1) взяти відрізок s ;
- 2) показати відрізок s ;
- 3) покласти відрізок s рівним a, b ;
- 4) довжина відрізка s ;
- 5) чи лежить точка t на одній прямій з відрізком s ;
- 6) чи лежить точка t всередині відрізка s ;

7) площа трикутника, утвореного точкою t та відрізком s .

У файлі записано послідовність точок. З використанням класів роботи з точками та відрізками на площині знайти:

а) трикутник з найбільшою площею, утворений точками послідовності;

б) коло найменшого радіуса, всередині якого лежать всі точки послідовності;

в) відрізок, на якому лежить найбільша кількість точок послідовності;

г) коло, на якому лежить найбільша кількість точок послідовності.

13.10. Клас Point (точка екрану) описано наступним чином:

```
import turtle

class Point:
    '''Точка екрану

    '''
    _count = 0

    def __init__(self, x, y):
        self._x = x      # _x - координата x точки
        self._y = y      # _y - координата y точки
        self._visible = False  # _visible - чи є точка
                                # видимою на екрані
        Point._count += 1

    def getx(self):
        '''Повертає координату x точки
        '''
        return self._x

    def gety(self):
        '''Повертає координату y точки
        '''
        return self._y

    def onscreen(self):
        '''Перевіряє, чи є точка видимою на екрані
        '''
        return self._visible

    def switchon(self):
```

```

'''Робить точку видимою на екрані
'''
if not self._visible:
    self._visible = True
    turtle.up()
    turtle.setpos(self._x, self._y)
    turtle.down()
    turtle.dot()

def switchoff(self):
'''Робить точку невидимою на екрані
'''
if self._visible:
    self._visible = False
    turtle.up()
    turtle.setpos(self._x, self._y)
    turtle.down()
    turtle.dot(turtle.bgcolor())

def move(self, dx, dy):
'''Пересуває точку на екрані на dx, dy позицій
'''
vis = self._visible
if vis:
    self.switchoff()
self._x += dx
self._y += dy
if vis:
    self.switchon()

def printcount():
    print('Кількість точок:', Point._count)

printcount = staticmethod(printcount)

```

Описати клас Прямокутник на базі класу Point. Сторони прямокутника паралельні осям координат. Для прямокутника задані лівий верхній кут та довжини сторін. Описати методи отримання довжини кожної з сторін, стерти прямокутник з екрану та зробити прямокутник видимим на екрані.

Скласти програму створення заданої кількості прямокутників та переміщення їх по екрану.

Вказівка. Використати графічну бібліотеку turtle.

13.11. Клас Point (точка екрану) описано наступним чином (див.

13.10.)

Описати клас Трикутник на базі класу Point. Основа трикутника паралельна осі x координат. Для трикутника задані лівий нижній кут (координати) та довжини сторін. Описати методи отримання довжини кожної з сторін, стерти трикутник з екрану та зробити трикутник видимим на екрані.

Скласти програму створення заданої кількості трикутників та переміщення їх по екрану.

Вказівка. Використати графічну бібліотеку turtle.

13.12. Клас Point2 описано наступним чином:

```
class Point2:
    """Клас реалізує точку площини"""
    def __init__(self, x, y):
        self._x = x
        self._y = y

    def get_x(self):
        """Повернути координату x"""
        return self._x

    def get_y(self):
        """Повернути координату y"""
        return self._y

    def __str__(self):
        """Повернути рядок представлення точки"""
        return "({}, {})".format(self._x, self._y)
```

Описати клас Rectangle – прямокутник, заданий 4 вершинами. Реалізувати методи повернення вершин, обчислення периметру та площі, __str__.

Використати цей клас для розв'язання задачі. Задано декілька прямокутників, сторони яких паралельні осям координат.

Знайти площу

- а) перетину цих прямокутників;
- б) об'єднання цих прямокутників.

13.13. Клас Point2 описано у задачі 13.12. Описати клас RegularPolygone - правильний багатокутник. Правильний багатокутник задається списком вершин (точок). Реалізувати

методи повернення вершин, обчислення периметру та площі, `__str__`.

Використати цей клас для розв'язання задачі. Задано декілька правильних багатокутників. Перевірити, чи є серед них такий, який охоплює всі інші багатокутники (тобто всі вершини всіх інших багатокутників лежать всередині цього багатокутника).

13.14. Використати стандартний контейнер Counter для розв'язання задачі. Дана непорожня послідовність символів (рядок) S . Описати функцію, яка визначає загальну кількість цифр та знаків операцій $+$, $-$, $*$, $/$, що входять до рядка S .

13.15. Використати стандартний контейнер Counter для розв'язання задачі.

а) знайти символ, який входить у рядок S максимальну кількість разів;

б) перевірити, чи складаються рядки $S1$, $S2$ з одних і тих же символів, які входять у ці рядки однаково кількість разів;

в) перевірити, чи вірно, що всі символи рядка $S1$, входять також у рядок $S2$, причому не меншу кількість разів, ніж у $S1$;

г) перевірити, чи вірно, що жоден символ рядка $S1$ не входить у рядок $S2$, та жоден символ $S2$ не входить до $S1$.

13.16. Ненульові елементи розрідженої матриці зберігаються у словнику (defaultdict). Ключами словника є кортежі, що складаються з індексів рядка та стовпчика, а значеннями словника, - значення елементів матриці. Обчислити:

а) суму двох розріджених матриць однакового розміру

б) добуток двох розріджених матриць

в) значення мінімального елемента розрідженої матриці

г) значення максимального елемента розрідженої матриці

д) добуток розрідженої матриці на розріджений вектор (зберігається також у словнику)

е) чи є розріджена матриця нижньою трикутною

є) чи є розріджена матриця діагональною

13.17. За допомогою стандартного контейнера deque розв'язати задачу:

По колу розташовано n гравців з номерами від 1 до n . У лічильці m слів. Починають лічити з першого гравця. m -й за ліком вибуває. Потім знову лічать з наступного гравця за

вибулим. Знову m -й вибуває. Так продовжують, поки не залишиться жодного гравця. Треба показати послідовність номерів, що вибувають, при заданих n та m .

13.18. За допомогою стандартного контейнера `deque` розв'язати задачу:

У магазині стоїть черга з m покупців. Час обслуговування покупця з черги – це випадкове ціле число в діапазоні від 1 до t_1 . Час додавання нового покупця до черги - це випадкове ціле число в діапазоні від 1 до t_2 . Промоделювати стан черги (тобто показати час виникнення подій – обслуговування та додавання покупця) за період часу T ($T \gg t_1, T \gg t_2$). Показати залишок черги.

13.19. Розв'язати задачу 13.18, передбачивши що через випадковий час від 1 до t_3 до початку черги додається „пільговий” покупець, який обслуговується першим, а через випадковий час від 1 до t_4 не витримує та йде з черги останній покупець.

13.20. За допомогою стандартного контейнера `deque` розв'язати задачу:

Є невпорядкована послідовність дійсних чисел. Ці числа треба розмістити у контейнері `deque` та, щоб числа буди впорядковані за неспаданням, починаючи від початку `deque`.

`sort` або інші способи сортування списків не використовувати.

13.21. За допомогою стандартного контейнера `deque` розв'язати задачу:

Є невпорядкована послідовність дійсних чисел. Ці числа треба розмістити у контейнері `deque` так, щоб числа буди впорядковані за неспаданням, починаючи від початку `deque`.

`sort` або інші способи сортування списків не використовувати.

13.22. Клас `Point2` описано у задачі 13.12

Описати клас `Point2Ex` як нащадок `Point2`. У цьому класі реалізувати особливі методи для відношень `<`, `=` тощо. Вважати що точки впорядковані за відстанню від початку координат.

Використати клас Point2Ex для розв’язання задачі: знайти довжину ламаної лінії, яка проведена між послідовністю точок, які впорядковано за відношенням “<”.

13.23. Клас Segment, який використовує клас Point2, описано наступним чином

```
class Segment:
    """Клас реалізує відрізок на площині"""

    def __init__(self, a, b):
        self._a = a      # точка
        self._b = b      # точка

    def get_a(self):
        """Повернути точку a"""
        return self._a

    def get_b(self):
        """Повернути точку b"""
        return self._b

    def __str__(self):
        """Повернути рядок представлення відрізка"""
        return "[{}, {}]".format(self._a, self._b)

    def len(self):
        return sqrt((self._a.get_x() - self._b.get_x())
** 2 +
                    (self._a.get_y() - self._b.get_y())
** 2)
```

Описати клас SegmentEx як нащадок Segment. У цьому класі реалізувати особливі методи для відношень <, == тощо. Вважати що відрізки впорядковані за їх довжиною.

Ввести декілька відрізків та показати їх у порядку впорядкування.

13.24. Клас Triangle, який використовує клас Point2, описано наступним чином

```
class Triangle:
    """Клас реалізує трикутник"""

    def __init__(self, a, b, c):
        self._a = a      # точка
        self._b = b      # точка
        self._c = c      # точка
```

```

def get_a(self):
    """Повернути точку a"""
    return self._a

def get_b(self):
    """Повернути точку b"""
    return self._b

def get_c(self):
    """Повернути точку c"""
    return self._c

def __str__(self):
    """Повернути рядок представлення трикутника"""
    return "Трикутник ({}, {}, {})".format(self._a,
self._b, self._c)

def _get_sides(self):
    s1 = Segment(self._a, self._b).len()
    s2 = Segment(self._b, self._c).len()
    s3 = Segment(self._a, self._c).len()
    return s1, s2, s3

def perimeter(self):
    """Повернути периметр трикутника"""
    s1, s2, s3 = self._get_sides()
    return s1 + s2 + s3

def square(self):
    p = self.perimeter() / 2
    s1, s2, s3 = self._get_sides()
    return sqrt(p * (p - s1) * (p - s2) * (p - s3))

```

Описати клас `TriangleEx` як нащадок `Triangle`. У цьому класі реалізувати особливі методи для відношень `<`, `==` тощо. Вважати що трикутники впорядковані за їх площею.

Ввести декілька трикутників та показати їх у порядку впорядкування.

13.25. Описати клас для роботи з відрізками на числовій осі. Для відрізка передбачити поля:

(a, b, empty)

де *a, b* - границі відрізка, *empty* - ознака того, що відрізок порожній.

Реалізувати методи:

- 1) зробити відрізок t порожнім;
- 2) чи порожній відрізок t ;
- 3) покласти відрізок t рівним a, b ;
- 4) покласти відрізок t рівним перетину відрізків $t1, t2$;
- 5) описати особливий метод для відношення `in` (`__contains__(self, other)`), який перевіряє, чи містить один відрізок інший.
- 6) реалізувати особливі методи для відношень `<`, `==` тощо. Вважати що відрізки впорядковані за їх довжиною.

З використанням класу скласти програму, яка вводить декілька відрізків та перевіряє, чи існує відрізок, який містить усі інші.

13.26. Описати клас для реалізації мультимножини цілих чисел на базі словника. Мультимножина - це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість входжень.

Кількість входжень елемента k ($1 \leq k \leq n$) у мультимножину - це значення елемента словника з ключем k .

Реалізувати методи:

- 1) зробити мультимножину порожньою;
- 2) чи є мультимножина порожньою;
- 3) додати елемент до мультимножини;
- 4) забрати елемент з мультимножини (кількість входжень елемента зменшується на 1, якщо елемент не входить - відмова);
- 5) кількість входжень елемента у мультимножину - реалізувати особливий метод `__getitem__`;
- 6) об'єднання двох мультимножин (в результаті об'єднання кількість входжень елемента визначається як максимальна з двох мультимножин) - реалізувати особливий метод для операції | (вертикальна риска) `__or__(self, other)`;
- 7) перетин двох мультимножин (в результаті кількість входжень елемента визначається як мінімальна з

двох мультимножин) – реалізувати особливий метод для операції & `__and__(self, other)`;

З використанням класу розв’язати задачі:

а) знайти символ, який входить у рядок S максимальну кількість разів;

б) перевірити, чи складаються рядки $S1$, $S2$ з одних і тих же символів, які входять у ці рядки однаковою кількістю разів;

в) перевірити, чи вірно, що всі символи рядка $S1$, входять також у рядок $S2$, причому не меншу кількість разів, ніж у $S1$.

13.27. В умовах задачі 13.26 реалізувати у класі для мультимножин особливі методи для відношень $<$, $=$ тощо. Вважати що відношення $<$ позначає включення однієї мультимножини у іншу.

З використанням класу розв’язати задачу. Ввести послідовність мультимножин та перевірити, чи є вона лінійно впорядкованою за відношенням “ $<$ ”

13.28. Клас `Point2` описано у задачі 13.12

Описати клас `Point2Ex` як нащадок `Point2`. У цьому класі реалізувати особливі методи для відношень $=$, $!=$. Вважати що точки рівні, якщо рівні їх координати.

Клас `Segment`, який використовує клас `Point2`, описано у задачі 13.26

Описати клас `SegmentEx`, як нащадок `Segment`. У цьому класі реалізувати особливі методи для відношень $<$, $=$ тощо. Вважати що відрізки впорядковані за довжиною. Також реалізувати метод, що перевіряє, чи лежить відрізок на одній прямій з іншим відрізком.

Використати класи `Point2Ex`, `SegmentEx` для розв’язання задачі: задано декілька відрізків. Перевірити, чи складають вони багатокутник та чи є він правильним.

13.29. На вхід надходить послідовність точок площини. Провести між ними ламану лінію найменшої довжини.

Для розв’язання задачі описати клас `Polyline` – ламана лінія.

Використати класи `Point2` та `Segment`, описані у задачах 13.12, 13.23.

У класі `Polyline` описати методи:

Конструктор `__init__`

add(p) – додати точку p до ламаної лінії

len() - довжина ламаної лінії

__le__ - чи менше одна ламана лінія іншої за довжиною

__str__ - відображення точок ламаної лінії у вигляді рядка

13.30. Клас Point (точка екрану) описано наступним чином (див. 13.10.)

Описати клас ПравильнийМногокутник (RegularPolygone) на базі класу Point (для многокутника визначено точку – центр многокутника, кількість сторін, а також довжину сторони).

Використати цей клас для створення та переміщення многокутників на екрані. Також розв’язати наступну задачу: на екрані декілька правильних многокутників з різною кількістю сторін. З’ясувати, який з них має максимальну площу.

13.31. Опишіть клас QuadraticEquation (Квадратне рівняння).

- Продумайте які атрибути та методи повинен мати клас.
- Реалізуйте конструктор копіювання для екземплярів класу.
- Реалізуйте метод, що повертає розв’язки цього рівняння.

Для заданого набору квадратних рівнянь, покажіть ті з них, що:

- не мають розв’язків;
- мають один розв’язок;
- мають два розв’язки;
- мають нескінченну кількість розв’язків.

Крім цього, з усіх рівнянь, що мають рівно один розв’язок вкажіть те, що має

- найменший розв’язок;
- найбільший розв’язок.

Для виведення рівняння опишіть метод show(), що виводить рівняння на екран у вигляді зручному для розуміння.

Вказівка. Всі рівняння задаються у текстовому файлі, як набір коефіцієнтів рівняння, записаних у порядку спадання відповідних степенів.

13.32. Опишіть класи для таких геометричних фігур зазначених нижче:

- Трикутник – визначається довжинами трьох сторін.
- Прямокутник – визначається двома сторонами
- Трапеція – визначається двома основами та двома бічними сторонами.
- Паралелограм – визначається двома сторонами та висотою.
- Круг – визначається радіусом.

У кожному з класів реалізуйте операції знаходження периметра (для кола – довжини кола) та площі. За допомогою цих класів розв'яжіть таку задачу: Задано список фігур вищенаведених класів. Серед заданих фігур знайдіть фігуру, що має найбільшу площу та периметр.

Перелік фігур зберігається у текстовому файлі – у кожному окремому рядку файла вказується назва фігури та список параметрів, що визначають фігуру відповідно до зазначеного вище. Параметри розділені одним або кількома символами пропуску. Назви фігур вказані таким чином: Triangle – Трикутник, Rectangle – Прямокутник, Trapeze – Трапеція, Parallelogram – Паралелограм, Circle – Круг.

13.33. Опишіть клас Vector для моделювання n-вимірних векторів. Реалізуйте методи

- Конструктор (з підтримкою копіювання).
- Виведення вектора на екран.
- Визначення розмірності вектора.
- Визначення довжини вектора.
- Визначення середнього арифметичного його компонент.
- Визначення найбільшої/найменшої компоненти вектора.

Серед заданого набору векторів, знайдіть:

- а) Той з них, що має найбільшу розмірність. Якщо таких векторів кілька, виведіть той з них, що має найменшу довжину.
- б) Вектор, що має найбільшу довжину. Якщо серед набору міститься кілька таких векторів, покажіть той з них, що має найменшу розмірність.
- в) Середню довжину вектору серед заданого набору.

г) Кількість векторів, що мають довжину більшу за середню довжину заданого набору.

д) Вектор з максимальною найбільшою компонентою. Якщо таких векторів у наборі міститься кілька, вкажіть той з них, що має меншу найменшу компоненту.

е) Вектор з мінімальною найменшою компонентою. Якщо таких векторів у наборі міститься кілька, вкажіть той з них, що має більшу найбільшу компоненту.

13.34. Опишіть клас для геометричних фігур зазначених нижче:

- 1) П'ятикутник;
- 2) Шестикутник;
- 3) Багатокутник (загальний).

У кожному з класів реалізуйте метод перевірки чи є цей багатокутник опуклим. За допомогою цих класів розв'яжіть таку задачу: Серед заданих багатокутників знайдіть всі які є опуклими.

Дані багатокутників мають зчитуватися з текстового файлу. Продумайте власний універсальний формат зображення даних у такому файлі.

13.35. Опишіть класи

- 1) Гість, що містить всю необхідну інформацію про жильця деякого готелю: ім'я, період проживання, кількість грошей виділених на проживання у готелі тощо.
- 2) Кімната містить інформацію про кімнату готелю у тому числі вартість проживання за добу її доступність у визначений період (інформацію про те ким і коли вони зайняті тощо).
- 3) Готель (клас-менеджер) – містить список кімнат цього готелю, є посередником між гостем і кімнатою, тобто забезпечує для нього відповідну кімнату, тощо.

Використовуючи вищенаведені класи розв'яжіть задачі:

- а) Про кількість вільних кімнат у готелі на вказаний момент;
- б) Пошуку вільної кімнати у зазначений період;
- в) Поселити жильця на вказаний термін;
- г) Вартості проживання жильця у зазначений період;
- д) Прибутку, який отримає готель за вказаний період;

е) Пошуку гостя у готелі (у заданий період);

13.36. Опишіть клас `Matrix2D`, що моделює роботу з квадратними матрицями розмірності 2. Опишіть методи введення (з клавіатури та з файлу) та виведення (на екран та у файл) матриці. Реалізуйте методи відшукування визначника та перевірки чи матриця є виродженою.

Опишіть клас `Vector2D` для 2-вимірних векторів.

Опишіть клас `Solver`, що використовує вищенаведені класи та розв'язує методом Крамера систему лінійних алгебраїчних рівнянь.

Використовуючи описані структури, напишіть програму, що розв'язує набір заданих алгебраїчних рівнянь.

13.37. Опишіть клас `Triangle` (Трикутник), для зображення трикутників на екрані за допомогою модуля `turtle`. Передбачте метод для встановлення кольору зображення трикутника. Використовуючи цей клас, зобразіть на екрані 100 трикутників. Розміри, позиції та кольори трикутників мають вибиратися випадковим чином.

Вказівка. Використайте модуль `random` генерації випадкових чисел:

```
from random import randint
# генерація випадкового числа з проміжку [0, 10]
rand = randint(0, 10)
```

13.38. Опишіть клас `Flower` (Квітка) для зображення квітки на екрані за допомогою модуля `turtle`. Продумайте структуру класу – квітка має бути композицією класів `Petal` (Пелюстка), `Leaf` (Листок) та `Stem` (Стебло). Використовуючи цей клас, зобразіть на екрані букет квітів.

13.39. За допомогою модуля `turtle`, намалюйте аналоговий годинник, стрілки якого будуть оновлюватися кожні `t` секунд відповідно до поточного часу.

Вказівка. Опишіть класи:

- 1) Цифра – призначений для зображення аналогової цифри.
- 2) Циферблат – контейнер для об'єктів класу Цифра – призначений для зображення циферблату годинника з усіма цифрами.
- 3) Стрілка – призначений для зображення стрілок годинника.

13.40. Опишіть клас `Equation` (Рівняння), що моделює лінійне

алгебраїчне рівняння виду

$$bx + c = 0.$$

Опишіть у ньому метод `solve()`, що повертає усі розв'язки цього рівняння, залежно від його коефіцієнтів. Створіть класи `QuadraticEquation` (Квадратне рівняння) та `BiQuadraticEquation` (Біквадратне рівняння), що є нащадками класу `Equation` та моделюють рівняння виду

$$ax^2 + bx + c = 0.$$

та

$$ax^4 + bx^2 + c = 0.$$

відповідно.

Опишіть для них метод `solve()`, який заміщує відповідний метод батьківського класу та коректно розв'язує кожне з цих рівнянь, повертаючи кортеж його розв'язків. Чи можна задати клас `BiQuadraticEquation`, як нащадок класу `QuadraticEquation`, та використати вже описані у ньому методи? Якщо так, то оформіть це саме таким чином.

Використовуючи вищенаведену ієрархію класів розв'яжіть таку задачу: Задано список рівнянь, кожне з яких визначається списком коефіцієнтів починаючи з коефіцієнта при найвищому степені.

Наприклад, лінійне рівняння $2x + 1 = 0$ задається таким списком [2, 1], квадратне рівняння $3x^2 + x + 6 = 0$ відповідно списком [3, 1, 6], а біквадратне $3x^4 + 5x^2 + 6 = 0$ відповідно списком [3, 0, 5, 0, 6].

Усі рівняння зберігається у текстовому файлі – список коефіцієнтів кожного рівняння розміщується в окремому рядку. Коефіцієнти розділені одним або кількома символами пропуску.

Серед заданих рівнянь знайдіть ті з них, що

- не мають розв'язків;
- мають один розв'язок;
- мають два розв'язки;
- мають три розв'язки;
- мають чотири розв'язки;
- мають нескінченну кількість розв'язків.

Крім цього вкажіть рівняння, що має

- найменший розв'язок,

- найбільший розв'язок, серед розв'язків усіх рівнянь, що мають рівно один розв'язок.
Вказівка. Для виведення рівняння опишіть віртуальний метод show(), що виводить рівняння на екран у вигляді зручному для розуміння.

Тема 14 Рекурсивні структури даних

.....

14.1. Описати клас для реалізації стеку на базі зв'язування об'єктів. Передбачити виконання дій над стеком:

- 1) почати роботу;
- 2) чи порожній стек;
- 3) вштовхнути елемент у стек;
- 4) виштовхнути елемент зі стеку.

Використовуючи цей клас, розв'язати задачу: на вхід поступає послідовність. Впорядкувати цю послідовність за зростанням.

Вказівка. Для впорядкування використати 2 стеки.

14.2. Використовуючи клас для реалізації стеку на базі зв'язування об'єктів (див. завдання 14.1), скласти підпрограми:

- а) довжина стеку (запобігти знищенню стеку під час обчислення його довжини);
- б) інверсія стеку.

14.3. Описати клас для реалізації черги на базі зв'язування об'єктів. Передбачити виконання дій над чергою:

- 1) почати роботу;
- 2) чи порожня черга;
- 3) додати елемент до кінця черги;
- 4) взяти елемент з початку черги.

Використовуючи цей клас, скласти підпрограму обчислення довжини черги. Запобігти знищенню черги після виклику підпрограми. Передбачити рекурсивний та нерекурсивний варіант.

14.4. Використовуючи клас для реалізації черги (див. завдання

14.3), розв'язати наступну задачу. У черзі є m чисел. Проводять n випробувань, в результаті кожного з яких отримують випадкові числа 0 або 1. Якщо отримано 0, то треба взяти елемент з початку черги та показати його на екрані. Якщо отримано 1, то ввести число з клавіатури та додати до кінця черги. Після завершення випробувань показати залишок черги.

Вказівка. Використати генератор випадкових чисел.

14.5. Використовуючи клас для реалізації черги (див. завдання 14.3), розв'язати наступну задачу. У магазині стоїть черга з m покупців. Час обслуговування покупця з черги – це випадкове ціле число в діапазоні від 1 до t_1 . Час додавання нового покупця до черги – це випадкове ціле число в діапазоні від 1 до t_2 . Промодельовати стан черги (тобто показати час виникнення подій – обслуговування та додавання покупця) за період часу T ($T \gg t_1$, $T \gg t_2$). Показати залишок черги.

14.6. Використовуючи клас для реалізації черги (див. завдання 14.3), скласти підпрограми:

- а) інверсія черги;
- б) забрати n елементів черги;
- в) конкатенація двох черг;
- г) порівняти 2 черги.

14.7. Описати клас для реалізації деку на базі списку. Передбачити виконання дій над деком:

- 1) почати роботу;
- 2) чи порожній дек;
- 3) додати елемент до початку деку;
- 4) взяти елемент з початку деку;
- 5) додати елемент до кінця деку;
- 6) взяти елемент з кінця деку.

Використовуючи цей клас, скласти підпрограми: обчислення довжини деку, копіювання деків. Запобігти знищенню деку після виклику підпрограми обчислення його довжини.

14.8. Використовуючи клас для реалізації деку (див. завдання 14.7), реалізувати стек на базі деку. Реалізувати стек на базі деку – означає описати клас та реалізувати дії над стеком викликами відповідних підпрограм, що реалізують дії над деком. Для реалізованого стеку розв'язати задачу інвертування вхідної

послідовності.

14.9. Використовуючи клас для реалізації деку (див. завдання 14.7), реалізувати чергу на базі деку (див. попереднє завдання). Для реалізованої черги розв'язати задачу обчислення довжини черги.

14.10. Використовуючи клас для реалізації деку (див. завдання 14.7), розв'язати задачу 14.8, передбачивши однак чотири можливих результати кожного випробування (випадкові числа від 0 до 3):

- 0 – взяти елемент з початку деку та показати його на екрані;
- 1 – ввести число з клавіатури та додати його до початку деку;
- 2 – взяти елемент з кінця деку та показати його на екрані;
- 3 – ввести число з клавіатури та додати його до кінця деку.

14.11. Використовуючи клас для реалізації деку (див. завдання 14.7), розв'язати задачу 14.5, передбачивши що через випадковий час від 1 до t_3 до початку черги додається „пільговий” покупець, який обслуговується першим, а через випадковий час від 1 до t_4 не витримує та йде з черги останній покупець.

14.12. Використовуючи клас для реалізації деку (див. завдання 14.7), скласти підпрограми:

- а) інверсія деку;
- б) конкатенація двох деків;
- в) порівняти 2 деки;
- г) забрати n елементів з початку деку;
- д) забрати n елементів з кінця деку;

14.13. Описати клас для реалізації кільцевого списку на базі зв'язування об'єктів. Передбачити виконання дій над списком:

- 1) Почати роботу.
- 2) Довжина списку.
- 3) Перейти до наступного елемента.
- 4) Поточний елемент.
- 5) Вставити елемент.
- 6) Видалити елемент.

Використовуючи цей клас, розв'язати задачі:

- а) “лічилка”;
- б) пошук у списку елемента, рівного заданому числу $Search(L, x)$ та присвоєння для списків $Let(L1, L2)$;

в) знайти послідовність рівних елементів списку, що йдуть підряд, максимальної довжини;

г) видалити із списку всі повторні входження елементів;

д) якщо, список складається з чисел, знайти пару елементів списку, різниця між якими є максимальною за абсолютною величиною для всіх пар елементів списку.

14.14. Використовуючи клас для реалізації кільцевого списку цілих чисел (див. завдання 14.13), скласти підпрограми:

а) *Change(L, n)* - замінити поточний елемент списку *L* числом *n*;

б) *Copy(L, m, n, L1)* - виділити з списку *L* *n* елементів, починаючи з елемента з номером *m* у новий список *L1*;

в) *Del(L, m, n)* - видалення *n* елементів списку *L*, починаючи з *m*-го, по відношенню до поточного, елемента кільцевого списку.

14.15. Клас для реалізації бінарного дерева описано наступним чином:

```
class Btree:
    '''Реалізує бінарне дерево. '''
    def __init__(self):
        '''Створити порожнє дерево. '''
        self._data = None #навантаження кореня дерева
        self._ls = None #лівий син
        self._rs = None #правий син

    def isempty(self):
        '''Чи порожнє дерево?. '''
        return self._data == None and self._ls == None
and self._rs == None

    def maketree(self, data, t1, t2):
        '''Створити дерево. Дані у корені - data, лівий
син - t1, правий син - t2 '''
        self._data = data
        self._ls = t1
        self._rs = t2

    def root(self):
        '''Корінь дерева. '''
        if self.isempty():
            print('root: Дерево порожнє')
            exit(1)
        return self._data
```



```

def leftson(self):
    '''Лівий син. '''
    if self.isempty():
        t = self
    else:
        t = self._ls
    return t

def rightson(self):
    '''Правий син. '''
    if self.isempty():
        t = self
    else:
        t = self._rs
    return t

def updateroot(self, data):
    '''Змінити корінь значенням data.'''
    if self.isempty(): #якщо дерево порожнє,
        # додати лівого та правого сина
        self._ls = Btree()
        self._rs = Btree()
    self._data = data

def updateleft(self, t):
    '''Змінити лівого сина значенням t.'''
    self._ls = t

def updateright(self, t):
    '''Змінити правого сина значенням t.'''
    self._rs = t

```

Використовуючи цей клас, розв'язати задачі:

а) виведення дерева на екран *Print(t)* з відображенням структури дерева;

б) пошуку у дереві елемента, рівного заданому *Search(t, x)*;

в) побудови бінарного дерева пошуку та пошуку елемента у ньому (бінарне дерево називають деревом пошуку, якщо для будь-якого його піддерева корінь цього піддерева не менше кожної вершини лівого сина та не більше кожної вершини правого сина);

г) обчислення висоти дерева *Height(t)*;

д) перевірки, чи входить одне дерево у друге *IsIn(t1, t2)*;

е) перевірки, чи є два дерева рівними $IsEq(t1, t2)$;

є) перевірки, чи є бінарне дерево повним (бінарне дерево називають повним, якщо кожна його вершина, окрім, можливо, нижнього рівня – листів дерева – має двох синів);

ж) перевірки, чи є бінарне дерево ідеально збалансованим (бінарне дерево називають ідеально збалансованим, якщо для кожної його вершини кількість вершин у лівому та правому піддеревах розрізняється не більше, ніж на 1);

з) перевірки, чи є бінарне дерево збалансованим по висоті або AVL-деревом (бінарне дерево називають збалансованим по висоті або AVL-деревом, якщо для кожної його вершини висота лівого та правого піддерев розрізняється не більше, ніж на 1);

и) перевірки, чи є бінарне дерево деревом Фібоначчі (AVL-дерево називають деревом Фібоначчі, якщо виконуються правила: при висоті дерева h висота його лівого піддерева дорівнює $(h-1)$, а висота правого піддерева $(h-2)$; порожнє дерево та дерево з однією вершиною є деревами Фібоначчі; кожне піддерево дерева Фібоначчі також є деревом Фібоначчі).

14.16. Описати клас для реалізації сильно розгалуженого дерева. Вершину дерева представити у вигляді кортежу (<навантаження>, <список синів>). Передбачити виконання дій над деревом:

1) почати роботу;

2) чи порожнє дерево;

3) створити дерево з заданим навантаженням та списком синів;

4) повернути корінь дерева;

5) повернути список синів;

6) змінити корінь дерева;

7) змінити список синів.

Використовуючи цей клас, розв'язати задачі:

а) побудови бінарного дерева за сильно розгалуженим деревом;

б) пошуку у дереві елемента, рівного заданому числу $Search(t, x)$;

в) обчислення висоти дерева $Height(t)$;

г) перевірки, чи входить одне дерево у друге $IsIn(t1, t2)$.

14.17. Використовуючи клас із задачі 14.16, розв'язати наступну задачу. Навколо міста розташовані населені пункти, кожний з яких має унікальну назву. Карта населених пунктів та доріг утворює дерево з вершиною у місті. Жителі утворюють населені пункти, які розташовуються по дорозі від міста і шлях вимірюються в кілометрах від міста. По назві населеного пункту показати відстань до міста, а також населені пункти, які розташовані на аналогічній відстані.

14.18. Описати клас для реалізації орієнтованих графів на базі зв'язування об'єктів. Передбачити виконання дій над графом:

- 1). створити порожній граф;
- 2). вивести вершини графа;
- 3). вивести довжину графа;
- 4). повернути поточну вершину графа;
- 5). повернути список попередників даної вершини;
- 7). повернути список наступників даної вершини;
- 8). оновити дані по вершині;
- 9). оновити список попередників вершини;
- 10). оновити список наступників вершини;
- 11). видалити вершину;
- 12). оновити (додати) вершину.

Використовуючи цей клас, розв'язати задачі:

- а) перевірити, чи існує шлях між двома вершинами;
- б) знайти найкоротший шлях між двома вершинами.
- в) знайти найдовший шлях, що не є циклом та діаметр графу (довжина цього шляху)
- г) перевірити, чи є граф сильно зв'язним (граф є сильно зв'язним, якщо між будь-якими двома вершинами існує шлях).

14.19. Використовуючи клас для реалізації орієнтованих графів (див. завдання 14.18), побудувати підпрограми, що розв'язують задачі для орієнтованих графів:

- а) перевірити чи містить граф цикли
- б) знайти довжину найдовшого циклу
- в) побудувати список усіх дуг графу (дугу представити кортежем з двох вершин)
- г) побудувати список ізольованих вершин графу (з напівстепінню входу та виходу 0)

14.20. Скласти програму, що перевіряє правильність відступів у програмі у Python, що зберігається у деякому файлі. Відступ – це кількість символів пропуску у рядку перед першим символом, що не є пропуском.

Якщо у рядку поза областю коментаря є двокрапка ‘:’, то у наступному рядку відступ повинен бути більшим за поточний. Якщо у рядку немає двокрапки, то відступ у наступному рядку повинен дорівнювати поточному, або дорівнювати одному з попередніх відступів (менше поточного).

Вказівка: використати стек, у якому зберігати усі наявні відступи. Якщо з’являється новий відступ, - він додається до стеку. Якщо відступ зменшується, то більші відступи забираються зі стеку.

Тема 15 Обробка помилок та виключних ситуацій

.....

15.1. Скласти підпрограму та програму для обчислення значення натурального числа за заданим рядком символів, який є записом цього числа у системі числення за основою b ($2 \leq b \leq 16$). Використати функцію, яка за заданим символом повертає відповідну цифру у системі числення за основою b . Використати у цій функції твердження про стан програми `assert` для перевірки того, що відповідний символ є цифрою у системі числення за основою b . Обробити у підпрограмі помилку неправильного символу рядка та показати змістовне повідомлення про помилку.

15.2. Скласти функцію та програму для обчислення наближеного значення суми

$$y = \ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots, (|x| < 1).$$

Використати у цій функції твердження про стан програми `assert` для перевірки того, що параметр x відповідає заданій умові. Обробити у програмі помилку неправильного значення x та показати змістовне повідомлення про помилку.

15.3. Скласти функцію та програму для обчислення суми всіх доданків, модуль яких не менше $\varepsilon > 0$, у комплексній точці z

$$\arctg(z) = z - \frac{z^3}{3} + \frac{z^5}{5} - \dots + (-1)^n \frac{z^{2n+1}}{2n+1} + \dots, \quad (|z| < 1).$$

Використати у цій функції твердження про стан програми `assert` для перевірки того, що параметр z відповідає заданій умові. Обробити у програмі помилку неправильного значення z та показати змістовне повідомлення про помилку.

15.4. Скласти функцію зі змінною кількістю параметрів та програму для обчислення

$$f(x_1, \dots, x_n, y_1, \dots, y_n) = \sum_{i=1}^n x_i^2 + y_i^2 + x_i \cdot y_i$$

Вказівка: оформити x_i як позиційні, а y_i , - як ключові параметри.

Використати у цій функції твердження про стан програми `assert` для перевірки того, що кількість параметрів x_i дорівнює кількості параметрів y_i . Обробити у програмі помилку неправильної кількості параметрів та показати змістовне повідомлення про помилку.

15.5. Задані натуральне число i файл f , компоненти якого є цілими числами. Побудувати файл g , записавши в нього найбільше значення перших n компонент файлу f , потім наступних n компонент і т.д. Розглянути два випадки:

- а) число компонент файлу ділиться на n ;
- б) число компонент файлу не ділиться на n .

В цьому випадку остання компонента файлу g повинна дорівнювати найбільшій із компонент файлу f , які утворюють останню (неповну) групу.

Забезпечити обробку помилок при роботі з файлами.

15.6. У текстовому файлі записана непорожня послідовність дійсних чисел, які розділяються пропусками в одному рядку та можуть бути розташовані у різних рядках. Визначити функцію обчислення найбільшого з цих чисел.

Забезпечити обробку помилок, якщо у файлі зустрічаються не дійсні числа.

15.7. Описати клас Двохбайтне ціле число для роботи з цілими числами, представленими двома байтами. Інтервал

представлення при цьому – від -2^{15} (-32768) до $2^{15}-1$ (32767). Операції не можуть вивести за межі інтервалу представлення. Наприклад, $32767 + 1 == -32768$, $32767 + 2 == -32767$ і т.д. Якщо результат операції виводить за межі інтервалу представлення, повинна ініціюватися помилка переповнення.

Перевизначити у цьому класі операції +, -, *, //, %.

Описати також 3 класи обробки помилок для двохбайтних цілих чисел: загальний клас обробки помилок та два його підкласи для обробки помилки переповнення та помилки ділення на 0.

Використати цей клас для розв'язання задач:

а) обчислення $n!$

б) обчислення x^n , де x – ціле, n – невід'ємне ціле.

Забезпечити обробку помилок при виконанні обчислень.

15.8. Описати клас Поліном та реалізувати методи: введення поліному, виведення поліному, обчислення значення поліному у точці x , взяття похідної поліному, суми, різниці та добутку поліномів.

Описати також клас обробки помилок при неправильному введенні поліному (ступінь – не невід'ємне ціле число, коефіцієнт – не дійсне число) та забезпечити ініціювання помилки при неправильному введенні.

Використати цей клас для розв'язання задачі: ввести 2 поліноми P_1 , P_2 та рядок, який містить вираз, що залежить від 2 поліномів. Наприклад,

$$P_1 + P_2 * P_1 - P_2$$

Обчислити поліном, який буде значенням цього виразу.

Забезпечити обробку помилок неправильного введення поліному.

Вказівка: поліном представити у вигляді словника.

15.9. Описати клас для реалізації мультимножини на базі словника. Мультимножина - це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість входжень.

Кількість входжень елемента k ($0 \leq k \leq n$) у мультимножину - це значення елемента словника з ключем k .

Реалізувати дії над мультимножинами:

- 1) зробити мультимножину порожньою;
- 2) чи є мультимножина порожньою;
- 3) додати елемент до мультимножини;
- 4) забрати елемент з мультимножини (кількість входжень елемента зменшується на 1, якщо елемент не входить - відмова);
- 5) кількість входжень елемента у мультимножину;
- 6) об'єднання двох мультимножин (в результаті об'єднання кількість входжень елемента визначається як максимальна з двох мультимножин);
- 7) перетин двох мультимножин (в результаті кількість входжень елемента визначається як мінімальна з двох мультимножин);

Описати клас обробки помилки взяття елемента, який не входить до мультимножини.

З використанням класу розв'язати задачі:

а) перевірити, чи складаються рядки $S1$, $S2$ з одних і тих же символів, які входять у ці рядки однаковою кількістю разів;

б) перевірити, чи вірно, що всі символи рядка $S1$, входять також у рядок $S2$, причому не меншою кількістю разів, ніж у $S1$.

Забезпечити обробку помилок.

15.10. Скласти функцію, яка визначає, чи є заданий рядок правильним записом цілого числа (можливо зі знаком). Якщо рядок є правильним записом цілого числа, функція має повернути це число. Якщо рядок не є правильним записом цілого числа, функція має повернути None

Вказівка: Використати перетворення рядка до цілого числа та обробку помилки, що може виникнути.

За допомогою цієї функції розв'язати задачу. Дано рядок, у якому через пропуски записані цілі числа. Порахувати суму цих чисел.

15.11. Скласти функцію, яка повертає найдовше слово рядка. Слова розділені пропусками. Перевірити у функції тип її параметру. Якщо це не рядок – ініціювати помилку `TypeError`.

Скласти програму, яка перевіряє, як працює ця функція з параметром типу рядок та з параметром – числом.

15.12. Описати клас для реалізації стеку та клас обробки помилки взяття елемента порожнього стеку та отримання верхівки порожнього стеку. Передбачити виконання дій над стеком:

- 1) почати роботу;
- 2) чи порожній стек;
- 3) вштовхнути елемент у стек;
- 4) верхівка стеку;
- 5) забрати верхівку стеку.

Передбачити ініціювання виключення у разі взяття елемента порожнього стеку та отримання верхівки порожнього стеку.

Використовуючи цей клас, розв'язати задачу:

У стеку є n чисел. Проводять m випробувань, в результаті кожного з яких отримують випадкові числа 0 або 1. Якщо отримано 0, то треба отримати та показати верхівку стеку а також забрати верхівку стеку. Якщо отримано 1, то ввести число з клавіатури та вштовхнути у стек. Після завершення випробувань показати залишок стеку.

Забезпечити обробку помилки взяття елемента з порожнього стеку під час проведення випробувань.

Вказівка. Використати генератор випадкових чисел.

15.13. Описати клас для реалізації деку. Передбачити виконання дій над деком:

- 1) почати роботу;
- 2) чи порожній дек;
- 3) додати елемент до початку деку;
- 4) взяти елемент з початку деку;
- 5) додати елемент до кінця деку;
- 6) взяти елемент з кінця деку.

Описати клас обробки помилок взяття елемента з початку та з кінця порожнього деку.

Забезпечити ініціювання помилки у разі взяття елемента з початку та з кінця порожнього деку.

Використовуючи цей клас, розв'язати задачу 15.10, передбачивши однак чотири можливих результати кожного випробування (випадкові числа від 0 до 3):

0 – взяти елемент з початку деку та показати його на екрані;

- 1 – ввести число з клавіатури та додати його до початку деку;
- 2 – взяти елемент з кінця деку та показати його на екрані;
- 3 – ввести число з клавіатури та додати його до кінця деку.

Забезпечити обробку помилки взяття елемента з порожнього деку під час проведення випробувань.

15.14. Використовуючи клас для реалізації деку та клас обробки помилок деку (див. завдання 15.11), розв’язати задачу:

У магазині стоїть черга з m покупців. Час обслуговування покупця з черги – це випадкове ціле число в діапазоні від 1 до t_1 . Час додавання нового покупця до черги – це випадкове ціле число в діапазоні від 1 до t_2 . Через випадковий час від 1 до t_3 до початку черги додається „пільговий” покупець, який обслуговується першим, а через випадковий час від 1 до t_4 не витримує та йде з черги останній покупець. Промодельовати стан черги (тобто показати час виникнення подій – обслуговування та додавання покупця) за період часу T ($T > t_1$, $T > t_2$, $T > t_3$, $T > t_4$). Показати залишок черги покупців.

Передбачити обробку помилки взяття елемента з порожнього деку (при обслуговуванні покупця, якщо черга порожня).

15.15. Використовуючи клас для реалізації деку та клас обробки помилок деку (див. завдання 15.11), скласти підпрограми:

- а) забрати n елементів з початку деку;
- б) забрати n елементів з кінця деку;

Передбачити обробку помилки взяття елемента з порожнього деку.

15.16. Описати клас для реалізації кільцевого списку та клас обробки помилок отримання поточного елемента та видалення елемента порожнього списку. Передбачити виконання дій над списком:

- 1) почати роботу;
- 2) довжина списку;
- 3) перейти до наступного елемента;
- 4) поточний елемент;
- 5) вставити елемент;
- б) видалити елемент.

Використовуючи цей клас, скласти підпрограми:

а) $Change(L, n)$ - замінити поточний елемент списку L числом n ;

б) $Copy(L, m, n, L1)$ - виділити з списку L n елементів, починаючи з елемента з номером m у новий список $L1$;

в) $Del(L, m, n)$ - видалення n елементів списку L , починаючи з m -го, по відношенню до поточного, елемента кільцевого списку.

Передбачити обробку помилок отримання поточного елемента та видалення елемента порожнього списку.

15.17. Описати клас для реалізації орієнтованих графів. Передбачити виконання дій над графом:

- 1) створити порожній граф;
- 2) вивести вершини графа;
- 3) довжина графа;
- 4) повернути вершину;
- 5) повернути дані вершини;
- 6) повернути список попередників;
- 7) повернути список наступників;
- 8) оновити дані вершини;
- 9) оновити список попередників;
- 10) оновити список наступників;
- 11) видалити вершину;
- 12) оновити (додати) вершину

Описати 4 класи, що утворюють ієрархію класів обробки помилок у графах: батьківський клас (помилка у графі), помилка отримання/зміни навантаження (або списку попередників, списку наступників) для неіснуючої вершини, неправильні параметри підпрограми «Оновити (додати) вершину», невалідне посилання на неіснуючу вершину при додаванні/оновленні вершини.

Використовуючи ці класи, розв'язати задачі:

- а) перевірити, чи існує шлях між двома вершинами;
 - б) знайти найкоротший шлях між двома вершинами.
 - в) знайти найдовший шлях, що не є циклом та діаметр графу (довжина цього шляху)
 - г) перевірити, чи є граф сильно зв'язним (граф є сильно зв'язним, якщо між будь-якими двома вершинами існує шлях).
- Забезпечити обробку помилок під час дій над графами.

15.18. Скласти функцію та програму для обчислення суми всіх доданків, модуль яких не менше $\varepsilon > 0$, у комплексній точці z

$$\ln(1+z) = z - \frac{z^2}{2} + \frac{z^3}{3} - \dots + (-1)^{n+1} \frac{z^n}{n} + \dots \quad (|z| < 1);$$

Використати у цій функції твердження про стан програми `assert` для перевірки того, що параметр z відповідає заданій умові. Обробити у програмі помилку неправильного значення z та показати змістовне повідомлення про помилку.

15.19. Класи `Point2`, `Segment`, `Triangle` описані у задачах 13.12, 13.23, 13.24.

Описати клас `TriangleEx` – нащадок `Triangle`, у якому реалізувати класовий метод `fromsegment(cls, s, p)` побудови трикутника за даним відрізком s та даною точкою p . Описати також клас помилки невалідного трикутника. Забезпечити ініціювання помилки у методі `fromsegment`, якщо трикутник невалідний.

З використанням класу `TriangleEx` розв'язати задачу. Дано відрізок та послідовність точок. Побудувати на цьому відрізку трикутник найбільшої площі з використанням однієї з точок. Виконати обробку помилки невалідного трикутника з виведенням повідомлення та продовженням роботи.

15.20. Задано рядок, серед елементів якого містяться цифри. Використовуючи для всіх символів рядка функцію `int` перетворення символу у число, обчислити суму цифр заданого рядка.

15.21. Дано список чисел. Не використовуючи функцію `len` визначення кількості елементів у цьому списку та не використовуючи цикл по колекції, визначити:

- кількість елементів у списку;
- суму елементів у списку;
- значення найбільшого відношення (частки) серед елементів.

15.22. До програми з клавіатури надходить послідовність цифр. Послідовність задається доти, доки користувач не введе слово «досить». Слід зауважити, що користувач не є дисциплінованим і може замість цифр вводити будь-що. Якщо

користувач вводить з клавіатури число більше за 9, то програма ініціює виключення `RuntimeError`. Якщо користувач вводить число менше за 0, то програма ініціює виключення `TypeError`. Якщо користувач вводить дійсне значення з діапазону від 0 до 9, то програма ініціює виключення `ValueError`. Підрахувати кількість виключень кожного типу, що виникають у програмі.

15.23. У папці з програмою міститься набір текстових файлів, кожен з яких містить набір дійсних чисел. Крім цього, у цій же папці міститься файл `content.txt`, що містить перелік імен файлів.

Реалізуйте програму, що:

- підсумовує дані, з файлів, що вказані у файлі `content.txt`;
- коректно опрацьовує ситуацію, у випадку якщо файл `content.txt` не існує за вказаним розташуванням або не доступний для читання та виводить на екран відповідне повідомлення;
- коректно опрацьовує ситуацію, у випадку якщо файл з переліку зазначеного у файлі `content.txt` не існує за вказаним розташуванням або не доступний для читання;
- коректно опрацьовує ситуацію, якщо файл з переліку зазначеного у файлі `content.txt` містить не лише дійсні числа.

15.24. Побудувати клас для роботи з бінарними файлами, в яких записані цілі числа. У класи визначені поля: ім'я файлу, кількість чисел у файлі. Реалізувати методи, введення чисел з консолі в файл, створення файлу з масиву чисел, виведення змісту файлу на консоль, повернути число за даним номером, додавання до файлу масиву чисел в кінець, видалення числа за даним номером. Забезпечити обробку помилок при роботі з файлами.

Створити відповідні виключення для проблем при створенні файлу, проблем при читанні з файлу, некоректних номерах чи кількості чисел.

15.25. Побудувати клас для роботи з текстовими файлами, в яких записані дійсні числа які розділяються пропусками в одному рядку та можуть бути розташовані у різних рядках. У класі визначені поля: ім'я файлу, кількість чисел у файлі, кількість рядків файлу. Реалізувати методи:

- 1) введення чисел з консолі в файл рядок за рядком,

- 2) створення файлу з двовимірною масиву чисел,
- 3) виведення змісту файлу на консоль, повернути число за даним номером,
- 4) додавання до файлу масиву чисел в кінець новим рядком,
- 5) видалення числа за даним номером рядку та місцем в ньому.

Створити відповідні виключення для обробки проблем при створенні файлу, проблем при читанні з файлу, некоректних номерів чи кількості чисел. Забезпечити обробку помилок, якщо у файлі, що читається, зустрічаються не дійсні числа.

15.26. Побудувати клас ГіперРядок, що працює як звичайний рядок, але у якості символів в ньому використовується інший рядок та також він має атрибут Роздільник(типу рядок) - що відокремлює в запису ці символи. Методи класу:

- 1) введення/виведення в/з консолі та в/з текстового файлу;
- 2) введення та заміна роздільника;
- 3) метод конкатенації (з додаванням між рядками роздільника);
- 4) довжина рядку;
- 5) злиття символів – тобто перетворення масиву символів на єдиний символ типу рядок;
- 6) доступ до даного символу за квадратним дужками;
- 7) видалення даного символу.

Забезпечити ініціювання помилки при неправильному введенні та роботі з рядками та роботі з файлами.

15.27. Побудувати клас для роботи з функціями. Клас дозволяє задавати інтервал де шукається корінь функції та створювати функцію від ступенів дійсних чисел та від функцій косинус, корінь та логарифм. Створити методи для обчислення значення за формулою лівих прямокутників, за формулою правих прямокутників, формулою середніх прямокутників, по формулі трапецій, по формулі Сімпсона (параболічних трапецій).

Реалізувати метод семплювання функції – задаються межі інтервалу та кількість семплів на інтервалі, обчислюються дискретні значення в даних точках і будується й виводиться

таблиця, що містить пари точка-значення. Забезпечити обробку помилок при заданні та роботі функцій.

Тема 16 Ітератори та генератори

.....

16.1. Побудувати ітератор, який проходить всі елементи діапазону від 1 до n , повертаючи тільки

- а) парні числа
- б) непарні числа

16.2. Побудувати ітератор, який проходить всі елементи діапазону від 1 до n у оберненому порядку, повертаючи тільки

- а) парні числа
- б) непарні числа

16.3. Побудувати ітератор, який проходить всі елементи послідовності `seq`, повертаючи тільки

- а) елементи з парними індексами
- б) елементи з непарними індексами

16.4. Побудувати ітератор, який проходить всі елементи послідовності `seq` у оберненому порядку, повертаючи тільки

- а) елементи з парними індексами
- б) елементи з непарними індексами

16.5. Описати клас-ітератор, який проходить всі елементи послідовності `seq`, повертаючи тільки

- а) елементи з парними індексами
- б) елементи з непарними індексами

16.6. Описати клас-ітератор, який проходить всі елементи послідовності `seq` у оберненому порядку, повертаючи тільки

- а) елементи з парними індексами
- б) елементи з непарними індексами

16.7. Описати клас-ітератор, який проходить всі елементи послідовності `seq`, повертаючи тільки непорожні елементи

16.8. Описати клас-ітератор, який проходить всі елементи послідовності `seq` у оберненому порядку, повертаючи тільки

непорожні елементи

16.9. Нехай елементи послідовності seq можна впорядкувати за зростанням (неспаданням). Описати клас-ітератор, який проходить всі елементи послідовності seq , повертаючи елементи у порядку

- а) зростання
- б) спадання

Сама послідовність seq повинна залишатись незмінною.

16.10. Описати клас-ітератор, який проходить всі символи рядка, повертаючи тільки символи української абетки.

16.11. Описати клас-ітератор, який повертає слова рядка (слова розділяються одним або декількома пропусками):

- а) всі слова у порядку їх слідування у рядку
- б) всі слова у оберненому порядку слідування
- в) всі слова у порядку зростання їх довжини
- г) всі слова у порядку спадання їх довжини
- д) тільки слова, які є симетричними рядками.

16.12. Описати генератор-вираз, що повертає елементи вектору з n дійсних компонент:

- а) з парними індексами
- б) з непарними індексами

16.13. Описати генератор-вираз, що повертає елементи вектору з n дійсних компонент у порядку спадання індексів:

- а) з парними індексами
- б) з непарними індексами

16.14. Описати генератор-вираз, що повертає всі елементи матриці $m \times n$ та обчислити:

- а) суму елементів матриці
- б) мінімальний елемент матриці
- в) максимальний елемент матриці

16.15. Описати генератор-вираз, що повертає всі елементи матриці $m \times n$ разом з їх індексами та обчислити:

- а) мінімальний елемент матриці та номери його рядка та стовпчика
- в) максимальний елемент матриці та номери його рядка та стовпчика

16.16. Описати генератор-функцію, що повертає всі елементи

послідовності $\{x_k\}$ дійсних чисел, заданої співвідношенням, та обчислити суму всіх елементів послідовності при заданому значенні x , що за абсолютною величиною не менші заданого $\varepsilon > 0$:

$$\text{а) } x_k = \frac{(-1)^k x^{2k+1}}{(2k+1)!} \quad (k \geq 0);$$

$$\text{б) } x_k = \frac{x^{2k}}{(2k)!} \quad (k \geq 0);$$

$$\text{в) } x_k = \frac{(-1)^k x^k}{k!} \quad (k \geq 0);$$

$$\text{г) } x_k = \frac{x^{2k+1}}{(2k+1)!} \quad (k \geq 0);$$

$$\text{д) } x_k = \frac{x^k}{k!} \quad (k \geq 0);$$

$$\text{е) } x_k = \frac{(-1)^k x^{2k}}{(2k)!} \quad (k \geq 0).$$

16.17. Описати генератор-функцію, що повертає всі доданки нескінченної суми дійсних чисел, заданої співвідношенням, та обчислити суму всіх доданків при заданому значенні x , що за абсолютною величиною не менші заданого $\varepsilon > 0$:

$$\text{а) } y = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots, \quad (|x| < 1);$$

$$\text{б) } y = \frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots, \quad (|x| < 1);$$

$$\text{в) } y = \ln\left(\frac{1+x}{1-x}\right) = 2 \cdot \left[x + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right], \quad (|x| < 1);$$

$$\text{г) } y = \frac{1}{(1+x)^2} = 1 - 2x + 3x^2 - \dots, \quad (|x| < 1);$$

$$\text{д) } y = \frac{1}{(1+x)^3} = 1 - \frac{2 \cdot 3}{2}x + \frac{3 \cdot 4}{2}x^2 - \frac{4 \cdot 5}{2}x^3 + \dots, \quad (|x| < 1);$$

$$\text{е) } y = \frac{1}{1+x^2} = 1 - x^2 + x^4 - x^6 + \dots, \quad (|x| < 1);$$

$$\text{ж) } y = \sqrt{1+x} = 1 + \frac{1}{2}x + \frac{1}{2 \cdot 4}x^2 - \frac{1 \cdot 3}{2 \cdot 4 \cdot 6}x^3 + \dots, \quad (|x| < 1);$$

$$\text{з) } y = \frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2}x + \frac{1 \cdot 3}{2 \cdot 4}x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^3 + \dots, \quad (|x| < 1);$$

$$\text{і) } y = \arcsin x = x + \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots, \quad (|x| < 1).$$

16.18. Для орієнтовного графу (див. задачу 14.17 з теми 14) описати клас-ітератор, який повертає всі вершини графу у порядку зростання їх напівстепені входу.

16.19. Для бінарного дерева (див. задачу 14.15 з теми 14)

описати клас-ітератор, який повертає всі вузли дерева у порядку обходу у ширину. Обхід бінарного дерева у ширину передбачає повернення кореня, потім всіх його синів, потім всіх синів синів і т.д.

16.20. Для сильно розгалуженого дерева (див. задачу 14.16 з теми 14) описати клас-ітератор, який повертає всі вузли дерева у порядку обходу у ширину. Обхід сильно розгалуженого дерева у ширину передбачає повернення кореня, потім всіх його синів, потім всіх синів синів і т.д.

16.21. Для сильно розгалуженого дерева (див. задачу 14.16 з теми 14) описати клас-ітератор, який повертає всі вузли дерева у порядку обходу у глибину. Обхід сильно розгалуженого дерева у глибину передбачає повернення кореня, потім застосування цього ж обходу для кожного з його синів.

16.22. Для кільцевого списку (див. задачу 14.13 з теми 14) описати клас-ітератор, який повертає всі елементи списку, починаючи з поточного елемента.

16.23. Для бінарного дерева (див. задачу 14.15 з теми 14) описати генератор-функцію, яка повертає всі вузли дерева у порядку обходу у ширину. Обхід бінарного дерева у ширину передбачає повернення кореня, потім всіх його синів, потім всіх синів синів і т.д.

16.24. Для сильно розгалуженого дерева (див. задачу 14.16 з теми 14) описати генератор-функцію, яка повертає всі вузли дерева у порядку обходу у ширину. Обхід сильно розгалуженого дерева у ширину передбачає повернення кореня, потім всіх його синів, потім всіх синів синів і т.д.

16.25. Для сильно розгалуженого дерева (див. задачу 14.16 з теми 14) описати генератор-функцію, яка повертає всі вузли дерева у порядку обходу у глибину. Обхід сильно розгалуженого дерева у глибину передбачає повернення кореня, потім застосування цього ж обходу для кожного з його синів.

16.26. Для кільцевого списку (див. задачу 14.13 з теми 14) описати генератор-функцію, яка повертає всі елементи списку, починаючи з поточного елемента.

16.27. Описати клас-ітератор, який повертає усі діагональні елементи квадратної матриці. Використовуючи цей клас,

обчислити суму діагональних елементів матриці.

16.28. Описати генератор-функцію, яка повертає усі діагональні елементи квадратної матриці. Використовуючи цю функцію, обчислити суму діагональних елементів матриці.

16.29. Описати клас-ітератор, який повертає з заданого рядка тільки цифри по 1 цифрі в порядку їх слідування.

16.30. Описати генератор-функцію, яка повертає з заданого рядка тільки цифри по 1 цифрі в порядку їх слідування.

16.31. Описати генератор-функцію, яка проходить всі елементи послідовності seq у оберненому порядку, повертаючи тільки елементи з непарними індексами

16.32. Описати генератор-функцію, яка проходить повертає усі кроки шахової фігури Кінь від поточної позиції. За допомогою цього генератора розв'язати задачу: знайти шлях фігури з поля s1 на поле s2.

16.33. Описати генератор-функцію, яка генерує послідовність усіх простих чисел.

За допомогою цієї функції отримати усі прості числа у діапазоні від m до n.

Тема 17 Декоратори

.....

17.1. Побудувати декоратор, який коригує результат функції, що повертає число, так, щоб цей результат завжди був більше 0. Виконати перевірку роботи декоратора для деякої функції f .

17.2. Побудувати декоратор, який коригує результат функції, що повертає число, так, щоб цей результат x був у межах заданих границь $a, b (a \leq x \leq b)$. Виконати перевірку роботи декоратора для деякої функції f .

17.3. Побудувати декоратор, який перевіряє, чи дорівнює кількість позиційних параметрів кількості ключових параметрів функції, що декорується. Якщо не дорівнює, то ініціює виключення. За допомогою декоратора розв'язати задачу: скласти підпрограму зі змінною кількістю параметрів для обчислення функції

$$f(x_1, \dots, x_n, y_1, \dots, y_n) = \left(x_1 + \frac{1}{y_1}\right) * \left(x_2 + \frac{1}{y_2}\right) * \dots * \left(x_n + \frac{1}{y_n}\right);$$

Оформивши x_i як позиційні, а y_i – як ключові параметри.

17.4. Побудувати декоратор, який перевіряє, чи є всі параметри декорованої функції рядками. Якщо ні, то ініціює виключення. Виконати перевірку роботи декоратора для функції, яка за декількома рядками повертає список рядків без повторів.

17.5. Побудувати декоратор, який перевіряє, чи належать усі параметри декорованої функції заданому типу (ім'я цього типу є параметром декоратора). Якщо ні, то ініціює виключення. Виконати перевірку роботи декоратора для функції, яка обчислює середнє значення декількох числових змінних.

17.6. Побудувати декоратор, який перевіряє, чи містить функція, що декорується, тільки позиційні параметри. Якщо ні, то ініціює виключення. За допомогою декоратора розв'язати задачу: скласти підпрограму зі змінною кількістю параметрів для обчислення функції

$$f(x_1, \dots, x_n) = \begin{cases} 1, & \max_i x_i > \sum_{i=1}^n x_i \\ \sum_{x_i > 0} x_i, & \text{в інших випадках} \end{cases}$$

17.7. Побудувати декоратор, який перевіряє, чи містить функція, що декорується, тільки ключові параметри. Якщо ні, то ініціює виключення. Виконати перевірку роботи декоратора для деякої функції f .

17.8. Побудувати декоратор, який перевіряє, чи містить функція, що декорується, тільки ключові параметри з ключами-рядками та значеннями-цілими числами. Якщо ні, то ініціює виключення. З використанням декоратора обчислити слово, що входить у рядок найчастіше (побудувати функцію, яка за декількома ключовими параметрами $\langle \text{слово} \rangle : \langle \text{кількість входжень} \rangle$ повертає слово, яке має найбільшу кількість входжень).

17.9. Побудувати декоратор, який виконує збереження результатів виклику деякої функції, що залежить від одного натурального аргументу у списку (кешування результатів).

Параметром декоратора буде найбільше можливе значення аргументу функції. Якщо виклик функції здійснюється для аргументу, для якого у списку вже є результат, він повертається із списку. Інакше здійснюється виклик функції, а її результат додається до списку. Використати цей декоратор для рекурсивної функції, яка обчислює n -те число Фібоначчі.

Вказівка: описати список з n елементів як нелокальну змінну (nonlocal) у функції-декораторі. Спочатку встановити значення всіх елементів списку у None.

17.10. Побудувати декоратор, який виконує збереження результатів виклику деякої функції, що залежить від декількох позиційних аргументів у словнику (кешування результатів). Ключами будуть кортежі із значень аргументів, а значеннями – значення функції. Якщо виклик функції здійснюється для аргументів, для яких у словнику вже є результат, він повертається із словника. Інакше здійснюється виклик функції, а її результат додається до словника. Використати цей декоратор для функції, яка перевіряє, чи є задане натуральне число простим та визначити щільність простих чисел на проміжку від 1 до n методом Монте-Карло. Метод Монте-Карло полягає у проведенні m випадкових випробувань ($m \geq n$). Результати кожного випробування запам'ятовуються, а по закінченні доля випробувань, що мають істинне значення від всіх випробувань дає наближене значення ймовірності.

Вказівка: описати словник як нелокальну змінну (nonlocal) у функції-декораторі.

17.11. Побудувати декоратор, який здійснює трасування рекурсивної функції, показуючи номер рекурсивного виклику (глибину вкладень викликів) та значення параметрів перед викликом функції, а також номер рекурсивного виклику та результат після виклику функції. З використанням декоратора виконати трасування рекурсивних функцій для обчислення факторіалу та чисел Фібоначчі.

Вказівка: описати глибину вкладення викликів як нелокальну змінну (nonlocal) у функції-декораторі.

17.12. Нехай результатом функції f є матриця у вигляді списку списків. Побудувати декоратор, який перевіряє, чи є ця матриця

розрідженою. Якщо так, то замість списку списків повертає словник, що містить тільки ненульові елементи матриці. Якщо ні, то повертає результат функції. Будемо вважати, що матриця є розрідженою, якщо кількість ненульових елементів менше 0.1 кількості всіх елементів. Перевірити роботу цього декоратора для функції додавання двох матриць.

17.13. Нехай аргументами функції f є дві матриці. Функція сприймає матриці у вигляді списку списків. Кожним з двох фактичних параметрів може бути матриця у вигляді списку списків або у вигляді словника, що містить ненульові елементи матриці (для розрідженої матриці). Побудувати декоратор, який приводить обидва параметри до вигляду матриці як списку списків. Перевірити роботу цього декоратора для функції додавання двох матриць.

17.14. Нехай результатом функції f є список деяких елементів. Побудувати декоратор, який модифікує цей список так, щоб він не містив повторів. Перевірити роботу декоратора для функції, яка повертає список слів, що містяться у текстовому файлі.

17.15. Нехай функція f має декілька однотипних аргументів. Побудувати декоратор, який перемішує аргументи функції та подає їх на вхід функції у випадковому порядку. Перевірити роботу цього декоратора для функції побудови бінарного дерева пошуку (див. тему «Рекурсивні структури даних»).

17.16. Нехай функція f не повертає жодного результату, а друкує деякий рядок. Побудувати декоратор, який перед рядком показує його номер (номер виклику функції, що декорується). Перевірити роботу цього декоратора для функції виведення всіх рядків текстового файлу.

17.17. Побудувати декоратор, який показує кількість позиційних параметрів декорованої функції.

Застосувати цей декоратор до функції, яка обчислює середнє значення своїх аргументів (вважати, що всі аргументи – числа).

17.18. Є функція, яка обчислює частоту входжень символів у рядок. Ця функція повертає об'єкт класу Counter.

Треба обчислити частоту входжень символів у рядок без урахування регістру. Для цього побудувати декоратор, який переводить параметри декорованої функції, які є рядками, до

нижнього регістру.

17.19. Побудувати декоратор, який перевіряє, чи є результат декорованої функції рядком. Якщо так, то повертає інверсію цього рядка (рядок, записаний символами у оберненому порядку).

17.20. Описати декоратор з параметрами `typeschecker`, який перевіряє, що усі параметри декорованої функції мають типи, які вказані у параметрах декоратора `typeschecker` (декоратор має змінну кількість параметрів).

Якщо якийсь з параметрів функції не відповідає очікуваному типу, ініціювати виключення `TypeError` зі змістовним повідомленням про помилку.

Так само ініціювати виключення `TypeError`, якщо кількість аргументів функції не дорівнює кількості типів у параметрах декоратора.

Описати окрім декоратора довільну функцію для його перевірки, застосувати до неї декоратор та перевірити різні варіанти роботи декоратора (з ініціюванням виключень та без). Здійснити обробку помилок, щоб одночасно бачити різні варіанти роботи декоратора.

Тема 18 Множинне наслідування

.....

18.1. Описати класи `ХудожнійТвір`, `Техніка` та `Картина`. Клас `Картина` є нащадком класів `ХудожнійТвір` та `Техніка`.

Клас `ХудожнійТвір` має поля «назва твору», «рік написання», «автор», «жанр» та методи, створення, введення та виведення художнього твору.

Клас `Техніка` описує використану техніку живопису. Цей клас має поля «назва техніки» та «матеріал» та методи створення, введення та виведення.

Клас `Картина` має власні поля «ширина», «висота», «вартість» та методи створення, введення, виведення усіх характеристик

картини, а також методи читання та зміни вартості.

З використанням цих класів скласти програму проведення аукціону картин. На аукціон виноситься список картин. Є список покупців. В процесі аукціону покупці пропонують свою ціну. Перемагає покупець, що дав найбільшу ціну, якщо за визначений період часу жодний покупець не назвав більшої ціни.

18.2. Описати класи МузичнийТвір, Звукозапис та МузичнийЗапис. Клас МузичнийЗапис є нащадком класів МузичнийТвір та Звукозапис.

Клас МузичнийТвір має поля «назва твору», «рік створення», «автор», «жанр» та методи, створення, введення та виведення музичного твору.

Клас Звукозапис має поля «частота», «стандарт запису» та «тривалість» та методи створення, введення та виведення.

Клас МузичнийЗапис має власні поля «носії», «розташування» та методи створення, введення, виведення усіх характеристик музичного запису.

З використанням цих класів скласти програму побудови списку із загального списку музичних записів списку відтворення (плейлисту). Плейлист може бути побудований за жанром, за автором або у випадковому порядку. Може бути також вказана кількість творів або тривалість плейлисту.

18.3. Описати класи Жанр, Носій та Фільм. Клас Фільм є нащадком класів Жанр та Носій.

Клас Жанр має поля «назва жанру», «характерні особливості» та методи, створення, введення та виведення жанру.

Клас Носій має поля «тип носія», «кольорова гама» та «тривалість зберігання» та методи створення, введення та виведення.

Клас Фільм має власні поля «назва», «автор сценарію», «режисер», «тривалість» та методи створення, введення, виведення усіх характеристик фільму.

З використанням цих класів скласти програму опису сховища фільмів. Передбачити додавання фільму до сховища, виведення переліку фільмів за носіями, авторами сценаріїв, режисерами, часу до вичерпання терміну зберігання.

18.4. Описати класи Професія, РангДержслужби та

ДержавнаПосада. Клас ДержавнаПосада є нащадком класів Професія та РангДержслужби.

Клас Професія має поля «назва професії», «спеціальність за освітою» та методи, створення, введення та виведення професії.

Клас РангДержслужби має поля «номер рангу», «вимоги для отримання рангу» та методи створення, введення та виведення.

Клас ДержавнаПосада має власні поля «назва», «заробітна платня» та методи створення, введення, виведення усіх характеристик державної посади.

З використанням цих класів скласти програму проведення конкурсу на перелік державних посад. Програма повинна публікувати вимоги до конкурсантів, а також враховувати бали, отримані на різних етапах конкурсу.

18.5. Описати класи Текст, Шрифт та ТекстовийНадпис. Клас ТекстовийНадпис є нащадком класів Текст та Шрифт.

Клас Текст має поля «зміст тексту», «мова» та методи, створення, введення та виведення тексту.

Клас Шрифт має поля «назва шрифту», «кегель», «напівгрубий», «нахилений», «підкреслений» та методи створення, введення та виведення.

Клас ТекстовийНадпис має власне поле «колір» та методи створення, введення, виведення усіх характеристик текстового надпису.

З використанням цих класів скласти програму введення тексту з текстового файлу, отримання списку текстів, застосувавши мови до окремих рядків файлу, застосування заданих шрифтів до різних текстів та отримання списку текстових надписів. Показати результат.

18.6. Описати класи Вид, Літаюче та ЛітаючаІстота. Клас ЛітаючаІстота є нащадком класів Вид та Літаюче.

Клас Вид має поля «назва виду», «клас» та методи, створення, введення та виведення виду.

Клас Літаюче має поля «спосіб польоту», «кількість крил», «напрямок польоту», та методи створення, введення та виведення.

Клас ЛітаючаІстота має власне поле «назва» та методи створення, введення, виведення усіх характеристик літаючої істоти.

З використанням цих класів скласти програму введення списку літаючих істот та їх класифікації щодо видів, способів польоту, напрямів польоту.

18.7. Описати класи Матеріал, ФункціональнеПристосування та Меблі. Клас Меблі є нащадком класів Матеріал та ФункціональнеПристосування.

Клас Матеріал має поля «тип матеріалу», «назва матеріалу», «природність» та методи, створення, введення та виведення матеріалу.

Клас ФункціональнеПристосування має поля «назва пристосування», «вікові рекомендації» та методи створення, введення та виведення.

Клас Меблі має власні поля «тип», «назва» та методи створення, введення, виведення усіх характеристик меблів.

З використанням цих класів скласти програму введення каталогу меблів та їх класифікації щодо матеріалу, типу, функціонального призначення.

18.8. Описати клас Sorted, який здійснює повернення елементів об'єкту деякого класу у порядку неспадання (при цьому мається на увазі, що клас є таким, що ітерується, а елементи можуть бути впорядковані за неспаданням).

Використати описаний клас для реалізації множини, з якої елементи повертаються за неспаданням (SortedSet). Клас SortedSet повинен бути нащадком класів Sorted та set.

18.9. Описати клас-домішок TraceMixin, який здійснює трасування (виведення на екран) звернення до атрибутів деякого класу. Для цього можна використати спеціальний метод `__getattr__`.

Використати описаний клас для реалізації бінарного дерева з трасуванням звернення до атрибутів. Для цього описати клас TracedVtree, що є нащадком класів TraceMixin та Vtree (див. приклад до теми «Рекурсивні структури даних»).

З використанням класу TracedVtree побудувати бінарне дерево пошуку та показати порядок звернення до атрибутів класу.

18.10. Описати клас-домішок CompareMixin, який реалізує усі 6 стандартних відношень (`==`, `!=`, `<`, `>`, `<=`, `>=`) на базі одного реалізованого відношення `<` та бульових операцій. Для цього

треба реалізувати спеціальні методи `__eq__`, `__ne__` тощо.

Описати клас `Point2` (точка площини) з властивостями – координатами та методами створення точки та виведення точки.

Описати клас-нащадок `Point2`, який вводять порядок на точках (відношення `<`) та клас нащадок цього класу та домішку `CompareMixin`:

а) `XOrderPoint2` – впорядкування за першою координатою – та `FullXOrderPoint2` – всі відношення.

б) `YOrderPoint2` – впорядкування за другою координатою – та `FullYOrderPoint2` – всі відношення.

в) `DistOrderPoint2` – впорядкування за відстанню від початку координат – та `FullDistOrderPoint2` – всі відношення.

З використанням цих класів ввести список точок та показати їх у порядку незростання.

18.11. Описати клас-домішок `CompareMixin` (див. задачу 18.10).

Вагою числового списку назвемо суму модулів його елементів.

Описати клас зважений список `WeightedList`, що є нащадком стандартного класу `list` та реалізує відношення `<` у відповідності з вагою списків.

Описати клас нащадок цього класу та домішку `CompareMixin` `FullOrderWeightedList` – зважений список з повним порядком.

Ввести список зважених списків з повним порядком та перевірити, чи є всі вони у сенсі заданого порядку (мають рівну вагу).

18.12. Розв’язати задачу 18.10 для точок простору. При цьому встановити порядок:

а) за першою координатою;

б) за другою координатою;

в) за третьою координатою;

г) за відстанню від початку координат.

18.13. Описати клас-домішок `SeqTxt`, який містить методи переведення числової послідовності у текст та тексту у числову послідовність цілих або дійсних чисел.

Описати клас `TxtList`, який є нащадком цього домішку та стандартного класу `list`.

Описати класи `TxtVector` (вектор з перетворенням у текст) та `TxtMatrix` (матриця з перетворенням у текст). Ці класи повинні, зокрема, містити методи збереження вектору та матриці у текстовий файл та читання вектору та матриці з текстового файлу. Крім того, клас `TxtMatrix` повинен містити список об'єктів класу `TxtVector`.

З використанням цих класів розв'язати задачі:

а) у двох текстових файлах містяться два вектори однакового розміру. Обчислити їх скалярний добуток.

б) у двох текстових файлах містяться вектор та матриця. Обчислити добуток вектору на матрицю та зберегти у новому текстовому файлі.

в) у двох текстових файлах містяться вектор та матриця. Обчислити добуток матриці на вектор та зберегти у новому текстовому файлі.

г) у двох текстових файлах містяться дві матриці. Обчислити їх добуток та зберегти у новому текстовому файлі.

18.14. Описати клас `ObjCounter`, який вміє рахувати кількість створених об'єктів та їх тип (використати статичні поля класу). Клас також має містити метод `print`, який показує кількість та тип створених об'єктів.

Клас `Point2` описано наступним чином:

```
class Point2:
    """Клас реалізує точку площини"""
    def __init__(self, x, y):
        self._x = x
        self._y = y

    def get_x(self):
        """Повернути координату x"""
        return self._x

    def get_y(self):
        """Повернути координату y"""
        return self._y

    def __str__(self):
        """Повернути рядок представлення точки"""
        return "({}, {})".format(self._x, self._y)
```

Описати клас Point2ObjCounter, який є нащадком класів Point2 та ObjCounter, та рахує кількість створених точок.

Вести послідовність координат точок (координати точки вводяться в одному рядку через кому ',', введення порожнього рядка завершує введення послідовності) та показати кількість та тип створених об'єктів.

18.15. Вагою числового списку назвемо суму модулів його елементів.

Описати клас зважений список WeightedList, що є нащадком стандартного класу list та реалізує відношення <, ==, <=, >, >= != у відповідності з вагою списків (описати методи __lt__ тощо).

Ввести з клавіатури список зважених списків та перевірити, чи є всі вони рівними у сенсі заданого порядку (мають рівну вагу).

Тема 19 Метакласи та метапрограмування

.....

19.1. Абстрактний клас Drawable та його нащадок TurtleDraw
Описані наступним чином:

```
class Drawable(metaclass = ABCMeta):
    """Абстрактний клас для зображення точок та
    кіл заданих розмірів та кольору

    """

    @property
    @abstractmethod
    def color(self):
        """Властивість, що повертає/встановлює
        колір переднього плану."""
        pass

    @color.setter
```

```

    @abstractmethod
    def color(self, cl):
        pass

    @property
    @abstractmethod
    def bgcolor(self):
        """Властивість, що повертає/встановлює
колір фону."""
        pass

    @bgcolor.setter
    @abstractmethod
    def bgcolor(self, cl):
        pass

    @abstractmethod
    def draw_point(self, x, y, cl):
        """Зобразити точку з координатами x, y
кольором cl."""
        pass

    @abstractmethod
    def draw_circle(self, x, y, r, cl):
        """Зобразити коло з координатами центру x,
y радіусом r кольором cl."""
        pass

class TurtleDraw(Drawable):
    """Клас для зображення точок та кіл заданих
розмірів та кольору.

    TurtleDraw є нащадком абстрактного класу
Drawable та використовує засоби
роботи з графікою з модуля turtle.
    """

    def __init__(self):
        pause = 50
        turtle.up()
        turtle.home()

```

```

        turtle.delay(pause)

    @property
    def color(self):
        """Властивість, що повертає/встановлює
колір переднього плану."""
        return turtle.pencolor()

    @color.setter
    def color(self, cl):
        turtle.pencolor(cl)

    @property
    def bgcolor(self):
        """Властивість, що повертає/встановлює
колір фону."""
        return turtle.bgcolor()

    @bgcolor.setter
    def bgcolor(self, cl):
        turtle.bgcolor(cl)

    def draw_point(self, x, y, cl):
        """Зобразити точку з координатами x, y
кольором cl."""
        turtle.up()
        turtle.setpos(x, y)
        turtle.down()
        turtle.dot(cl)

    def draw_circle(self, x, y, r, cl):
        """Зобразити коло з координатами центру x,
у радіусом r кольором cl."""
        c = self.color
        self.color = cl
        turtle.up()
        turtle.setpos(x, y-r) #малює починаючи
знизу кола
        turtle.down()
        turtle.circle(r)
        self.color = c

```

Додати до Drawable (та TurtleDraw) метод для зображення прямокутників.

Описати клас(и) для гри у «хрестики-нолики» 3x3, який для показу поля використовує Drawable, та показати хід гри за допомогою TurtleDraw.

Описати власний клас-нащадок Drawable з ім'ям CharDraw, який зображує точки, кола та прямокутники у символічному режимі. Показати хід гри за допомогою CharDraw.

19.2. Абстрактний клас Drawable та його нащадок TurtleDraw описані наступним чином (див. завдання 19.1).

Додати до Drawable (та TurtleDraw) метод для зображення прямокутників.

Описати клас(и) для гри у «морський бій», який для показу поля використовує Drawable, та показати хід гри за допомогою TurtleDraw.

Описати власний клас-нащадок Drawable з ім'ям CharDraw, який зображує точки, кола та прямокутники у символічному режимі. Показати хід гри за допомогою CharDraw.

19.3. Описати абстрактний клас Shape (геометрична фігура) та його нащадки Circle (коло), Rectangle (прямокутник), Triangle (трикутник). Передбачити у Shape властивості обчислення периметру та площі фігури а також метод, що перевіряє, чи перетинаються дві фігури.

З використанням цих класів розв'язати задачу. Дано список геометричних фігур, що не перетинаються. Перевірити, чи справді вони не перетинаються та порахувати їх сумарну площу.

19.4. Описати абстрактний клас Creature (істота). Кожна істота має вік, термін життя, метод зростання протягом життя, спосіб живлення (інші істоти, ґрунт тощо), Необхідний об'єм живлення, швидкість та умови розмноження. Описати класи-нащадки Creature: Animal (тварина) та Herb (рослина). Описати декілька класів тварин та рослин, задати початкову конфігурацію (кількість істот кожного класу, їх вік) та промоделювати розвиток цієї конфігурації протягом часу T.

19.5. Описати декоратор класу, який модифікує клас для перевірки, чи належать параметри, що використовуються при ініціалізації об'єкта цього класу, заданим типам. Вважати, що

клас має поле `_field_types` – словник, що містить імена полів в якості ключів та типи полів в якості значень. Якщо типи невідповідні, то ініціює виключення `ValueError`.

Застосувати цей декоратор до класів `Point` та `Circle` (див. тему «Класи та об'єкти»), та виконати зображення та переміщення точок та кіл.

19.6. Описати декоратор класу, який здійснює модифікацію класу з метою трасування виклику усіх власних (не спеціальних) методів класу. Під час трасування показувати ім'я методу, значення параметрів до виклику, а також результат після виклику. Застосувати цей декоратор до класів `Person` та `Student` (див. тему «Класи та об'єкти») та виконати програму обчислення стипендії студентам.

19.7. Описати декоратор класу, який здійснює модифікацію класу з метою обчислення часу роботи усіх методів класу. Під час виклику методу показувати ім'я методу та час його роботи. Застосувати цей декоратор до класу `Btree` (див. приклад до теми «Рекурсивні структури даних») та побудувати бінарне дерево пошуку.

19.8. Описати декоратор класу, який здійснює модифікацію класу з метою перехоплення всіх виключень від усіх методів класу. При виникненні виключення у методі класу це виключення з усіма параметрами зберігається у текстовому файлі, а робота методу продовжується. Застосувати цей декоратор до класу `Queue` (див. приклад до теми «Обробка помилок та виключних ситуацій») та розв'язати наступну задачу. У магазині стоїть черга з m покупців. Час обслуговування покупця з черги – це випадкове ціле число в діапазоні від 1 до t_1 . Час додавання нового покупця до черги - це випадкове ціле число в діапазоні від 1 до t_2 . Промодельовати стан черги (тобто показати час виникнення подій – обслуговування та додавання покупця) за період часу T ($T \gg t_1$, $T \gg t_2$). Показати залишок черги.

Вказівка: підібрати параметри m , t_1 , t_2 так, щоб виникло виключення взяття елемента з порожньої черги.

19.9. Описати декоратор класу, який здійснює модифікацію класу з метою збереження у файлі та читання з файлу значень усіх полів об'єктів класу. Для цього декоратор повинен додати у клас

методи `save()` та `load()`. Застосувати цей декоратор до класу `Rlist` (див. приклад до теми «Рекурсивні структури даних») та реалізувати програму гри у відгадування слів зі збереженням та відновленням результатів гри.

19.10. Описати функцію, яка створює клас-нащадок `tuple`, що реалізує розмічене об'єднання.

Розмічене об'єднання – це тип даних, елементи якого можуть мати різний набір полів в залежності від значення поля-дискримінанта. Наприклад, точка площини може бути представлена у декартовій або полярній системі координат. Отже, поле дискримінанта може називатись `syst_coord` зі значеннями `'cart'`, `'polar'`, а відповідні поля у декартовій системі координат – `'x'`, `'y'`, а у полярній, – `'ro'`, `'phi'`. Доступ до полів розміченого об'єднання здійснюється наступним чином: `a.x` або `a.syst_coord`.

Параметрами функції, що реалізує розмічене об'єднання, будуть: ім'я класу, ім'я поля дискримінанта та словник з ключами, що є значеннями поля дискримінанта, та значеннями – наборами імен відповідних полів.

Побудований клас повинен забороняти доступ до полів, які відсутні (при заданому значенні поля дискримінанта).

Використавши цю функцію, побудувати клас для точки площини, що може бути задана у декартовій або полярній системі координат.

Нехай дано 3 точки площини у декартовій або полярній системі координат. Обчислити площу трикутника, утвореного цими точками.

19.11. Виконати задачу 19.10 з використанням метакласів. А саме, описати метаклас `UnionTupleMeta`, який створює класи розміченого об'єднання та клас `UnionTuple` – батьківський клас у ієрархії розмічених об'єднань.

19.12. Виконати задачу 19.9 з використанням метакласів замість декоратора. А саме, описати метаклас, який додає до створюваних ним класів `save()` та `load()`.

19.13. Виконати задачу 19.8 з використанням метакласів замість декоратора.

19.14. Виконати задачу 19.7 з використанням метакласів замість декоратора.

19.15. Виконати задачу 19.6 з використанням метакласів замість декоратора.

19.16. Описати декоратор класу `pool_deco`, який дозволяє відслідковувати, що кількість об'єктів певного класу не перевищує n . При спробі створити $(n+1)$ об'єкт, у якості результату повертається перший створений об'єкт, далі – другий тощо.

Використати цей декоратор для розв'язання задачі. Деякий клас `Incident` (частковий опис нижче) дозволяє зберігати та обробляти інформацію про інцидент (назва, тип, опис, час виникнення, заходи). Клас містить конструктор та методи `print` (відобразити інцидент на екрані), `save` – записати інцидент у текстовий файл.

Одночасно на екрані не може розміщуватись більш, ніж 9 інцидентів. Забезпечити одночасне знаходження у пам'яті тільки 9 інцидентів (в той же час інформація про всі інциденти повинна зберігатись у текстовому файлі).

Описати також реалізацію методів `__init__`, `print`, `save` для `Incident`

```
class Incident:
    def __new__(cls, time, name, type, actions):
        return object.__new__(cls)

    def __init__(self, time, name, type, actions):
        ...

    def print(self):
        ...

    def save(self):
        ...
```

19.17. Описати декоратор класу, який здійснює модифікацію класу з метою обчислення часу роботи усіх методів класу. Під

час виклику методу показувати ім'я методу та час його роботи. Застосувати цей декоратор до класу `Polynome` (див. приклад до теми «Наслідкування та агрегування») та обчислити суму 2 поліномів.

19.18. Описати декоратор класу, який здійснює модифікацію класу з метою перехоплення всіх виключень від усіх методів класу. При виникненні виключення у методі класу це виключення з усіма параметрами зберігається у текстовому файлі, а робота методу продовжується. Застосувати цей декоратор до класу `deque` та розв'язати наступну задачу. У магазині стоїть черга з m покупців. Час обслуговування покупця з черги – це випадкове ціле число в діапазоні від 1 до t_1 . Час додавання нового покупця до черги - це випадкове ціле число в діапазоні від 1 до t_2 . Промоделювати стан черги (тобто показати час виникнення подій – обслуговування та додавання покупця) за період часу T ($T \gg t_1$, $T \gg t_2$). Показати залишок черги.

Вказівка: підібрати параметри m , t_1 , t_2 так, щоб виникло виключення взяття елемента з порожнього деку.

19.19. Описати декоратор класу, який здійснює модифікацію класу з метою збереження у файлі та читання з файлу значень усіх полів об'єктів класу. Для цього декоратор повинен додати у клас методи `save()` та `load()`. Застосувати цей декоратор до класу `WordGuessGame` (див. приклад до теми «Множинне наслідкування») та реалізувати програму гри у відгадування слів зі збереженням та відновленням результатів гри.

19.20. Виконати задачу 19.19 з використанням метакласів замість декоратора. А саме, описати метаклас, який додає до створюваних ним класів `save()` та `load()`.

19.21. Виконати задачу 19.18 з використанням метакласів замість декоратора.

19.22. Виконати задачу 19.16 з використанням метакласу `PoolMeta` замість декоратора.

Підказка: у метакласі описати метод `__call__`.

19.23. Описати декоратор класу `ColumnPrint`, який модифікує метод `__str__` так, щоб той повертав для об'єкта класу його елементи у вигляді рядка у стовпчик (через `'\n'`), якщо об'єкт належить до типу, що ітерується (список, кортеж, словник, тощо).

Якщо ж об'єкт не належить до типу, що ітерується, - повернути той рядок, який повертає клас, що декорується.

19.24. Описати клас Messenger. Під час створення об'єкту цього класу передаються імена методів, які мають бути визначені для цього об'єкту.

Кожний метод при виклику має показувати своє ім'я та значення усіх своїх параметрів (print).

Підказка: описати та використати клас, який зберігає ім'я методу та спрацьовує при виклику методу.

Тема 20 Наукові обчислення (numru)

.....

20.1. Сформулювати гіпотези щодо наявності та значень границь послідовностей $\{a_n\}$. Для цього побудувати масиви numru зі значеннями n та a_n . Припустити, що границя послідовності $\{a_n\}$ дорівнює b . Для заданого малого $\varepsilon > 0$ перевірити, що у масиві a_n , починаючи з деякого k , для $m > k$ $|a_m - b| < \varepsilon$.

Графічно відобразити елементи послідовності, а також пряму $y = b$. Побудувати графік смуги $(b - \varepsilon, b + \varepsilon)$ та показати, що усі елементи a_n , при $n > k$, потрапляють у цю смугу. Самостійно підібрати масштаб осей для ілюстрації наявності границі.

Розглянути границі:

$$a) b = \lim_{n \rightarrow \infty} \frac{(n + \frac{n}{3} + \dots + \frac{n}{3^n})(n-3)\sqrt[n]{n}}{2n^2 + 5}$$

$$б) b = \lim_{n \rightarrow \infty} \frac{(n-1)^4 - (n+2)^4}{(2n+1)^3 + (n-1)^3}$$

$$\text{в) } b = \lim_{n \rightarrow \infty} \frac{n\sqrt[7]{n} + \sqrt[4]{16n^8} + 5}{(n + \sqrt[3]{n})\sqrt[5]{n^5 - 1}}$$

$$\text{г) } b = \lim_{n \rightarrow \infty} \frac{\sqrt{n(n^4 + 1)} - \sqrt{(n^3 - 1)(n^2 + 2)}}{\sqrt{n}}.$$

$$\text{д) } b = \lim_{n \rightarrow \infty} \frac{\sqrt[3]{2n^4 + n^3 + 1} - n\sqrt[3]{2n + 3}}{\sqrt[3]{n + 1}}.$$

$$\text{е) } b = \lim_{n \rightarrow \infty} \frac{(n + \frac{n}{3} + \dots + \frac{n}{3^n})(n - 3)\sqrt[n]{n}}{2n^2 + 5}.$$

$$\text{ж) } b = \lim_{n \rightarrow \infty} \left| \frac{1}{n} - \frac{2}{n} + \frac{3}{n} - \dots + \frac{(-1)^{n-1}n}{n} \right|.$$

$$\text{з) } b = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n$$

$$\text{и) } b = \lim_{n \rightarrow \infty} \left(\frac{5n^2 + 7n + 1}{5n^2 + 3n + 6} \right)^{n-3}.$$

$$\text{к) } b = \lim_{n \rightarrow \infty} 2n(\sqrt{n^2 + 1} - n)$$

$$\text{л) } b = \lim_{n \rightarrow \infty} \frac{n^\alpha}{a^n} \quad (\alpha > 0, a > 1)$$

$$\text{л) } b = \lim_{n \rightarrow \infty} \sqrt[n]{n}$$

$$\text{м) } b = \lim_{n \rightarrow \infty} \left(1 - \frac{2}{3} + \left(\frac{2}{3}\right)^2 - \left(\frac{2}{3}\right)^3 + \dots + (-1)^n \left(\frac{2}{3}\right)^n \right)$$

$$\text{н) } b = \lim_{n \rightarrow \infty} (\sqrt[3]{n^2 - n^3} + n)$$

$$\text{о) } b = \lim_{n \rightarrow \infty} \frac{\cos n - e^{n-1}}{e^n + \pi^{\frac{n}{2}}}$$

20.2. Виконати наближене обчислення π наближенням кола рівносторонніми n -кутниками. Для цього порахувати периметр рівностороннього n -кутника. Розглянути n -кутники з числом сторін $n = 2^k$, $k = 2, 3, 4, \dots$. Використати масиви numpy.

Зобразити на графіках коло та n -кутник. Зберегти відео (виконати анімацію) для різних значень k .

20.3. Виконати наближення функції $f(x)$ на відрізку $[a, b]$ частиною ряду Тейлора, що є розкладом $f(x)$ у 0. Взяти перші n доданків для $n = 1, 2, \dots, m$, де m – задане число. Побудувати графіки функції та її наближення. Зберегти відео (виконати анімацію) для різних значень n . Використати масиви numpy.

Розв'язати задачу для функції $f(x)$:

$$\text{а) } y = \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots;$$

$$\text{б) } y = \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots;$$

$$\text{в) } y = \sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots;$$

$$\text{г) } y = \cosh x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots;$$

$$\text{д) } y = e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots;$$

$$\text{е) } y = \ln(1+x) = x - \frac{x^2}{2!} + \frac{x^3}{3!} - \dots, \quad (|x| < 1);$$

$$\text{ж) } y = \frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots, \quad (|x| < 1);$$

$$\text{з) } y = \ln \frac{1+x}{1-x} = 2 * \left[\frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right], \quad (|x| < 1);$$

$$\text{и) } y = \frac{1}{(1+x)^2} = 1 - 2 * x + 3 * x^2 - \dots, \quad (|x| < 1);$$

$$\text{к) } y = \frac{1}{(1+x)^3} = 1 - \frac{2*3}{2} x + \frac{3*4}{2} x^2 - \frac{4*5}{2} x^3 + \dots, \quad (|x| < 1);$$

$$\text{л) } y = \frac{1}{1+x^2} = 1 - x^2 + x^4 - x^6 + \dots, \quad (|x| < 1);$$

$$\text{м) } y = \sqrt{1+x} = 1 + \frac{1}{2} x - \frac{1}{2*4} x^2 + \frac{1*3}{2*4*6} x^3 - \dots, \quad (|x| < 1);$$

$$\text{н) } y = \frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} x + \frac{1*3}{2*4} x^2 - \frac{1*3*5}{2*4*6} x^3 - \dots, \quad (|x| < 1);$$

$$\text{о) } y = \arcsin x = x + \frac{1}{2} \frac{x^3}{3!} + \frac{1*3}{2*4} \frac{x^5}{5!} + \dots, \quad (|x| < 1).$$

20.4. Виконати завдання 20.3 для заданої функції $f(x)$ на відрізку $[a, b]$ (варіанти а) – о)) без побудови відео (анімації). Взяти для наближення перші m членів ряду.

Замість відео на графіку функції та її наближення залити кольором ділянки неспівпадіння функції та наближення. Окрім цього, побудувати графік функції $g(x)$, яка набуває значення модуля різниці між функцією $f(x)$ та її наближенням.

Порахувати методом Монте-Карло середню похибку наближення як корінь відношення площі фігури (фігур) між кривими графіків функції на відрізку $[a, b]$ до площі охоплюючого прямокутника. В якості охоплюючого прямокутника взяти границі осей, які розраховує `matplotlib`.

20.5. Нехай ми маємо послідовність точок $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$. При цьому, $x_0 < x_1 < \dots < x_n$. Будемо вважати, що точки y_i є значеннями деякої функції f у точках x_i . Інтерполяцією називається побудова функції f у всіх точках на проміжку $[x_0, x_n]$.

Одним із способів інтерполяції є застосування інтерполяційного поліному Лагранжа, який будується за формулою:

$$P_L(x) = \sum_{k=0}^n y_k L_k(x), \quad \text{де } L_k(x) = \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}$$

Виконати наближення інтерполяційним поліномом Лагранжа функції $\sin(x)$ на відрізку $[0, 2\pi]$ у $(n+1)$ точці, де $n = 2^k$, $k = 2, 3, 4, \dots$ (скласти функцію для обчислення $P_L(x)$) Використати масиви numpy.

Зобразити на графіках функції $\sin(x)$ та $P_L(x)$. Зберегти відео (виконати анімацію) для різних значень k .

20.6. Виконати завдання 20.5 без побудови відео (анімації).

Замість відео на графіку функції та її наближення залити кольором ділянки неспівпадіння функції та наближення. Окрім цього, побудувати графік функції $g(x)$, яка набуває значення модуля різниці між функцією $\sin(x)$ та її наближенням.

Порахувати методом Монте-Карло середню похибку наближення як корінь відношення площі фігури (фігур) між кривими графіків функції на відрізку $[a, b]$ до площі охоплюючого прямокутника. В якості охоплюючого прямокутника взяти границі осей, які розраховує matplotlib.

20.7. В умовах завдань 20.5, 20.6, окрім інтерполяції поліномом Лагранжа, виконати також лінійну інтерполяцію функцією $\sin(x)$ у тих же n точках. Лінійна інтерполяція – це наближення функції відрізками прямих $[(x_i, y_i), (x_{i+1}, y_{i+1})]$.

На графіках функції та її наближення поліномом Лагранжа та лінійною інтерполяцією залити кольором ділянки неспівпадіння функції та наближення.

Порахувати методом Монте-Карло середню похибку наближення як корінь відношення площі фігури (фігур) між кривими графіків функції на відрізку $[a, b]$ до площі

охоплюючого прямокутника. В якості охоплюючого прямокутника взяти границі осей, які розраховує matplotlib.

Порівняти точність лінійної інтерполяції та інтерполяції поліномом Лагранжа.

20.8. Скласти програму перетворення дійсного вектору за наступним правилом: всі від'ємні компоненти вектору перенести до його початку, а всі інші - до кінця, зберігаючи початкове взаємне розташування як серед від'ємних, так і серед інших компонент. Використати масиви numpy та векторизувати програмний код.

20.9. Скласти програму обчислення норм дійсної матриці порядку n

$$\text{а) } \|A\| = \max_{i=1, \dots, n} \sum_{j=1}^n |a_{ij}|; \quad \text{б) } \|A\| = \max_{j=1, \dots, n} \sum_{i=1}^n |a_{ij}|;$$

Використати масиви numpy та векторизувати програмний код.

20.10. Скласти програму, яка перевіряє, чи є задана квадратна матриця з цілих чисел ортонормованою, тобто. такою, в якій скалярний добуток кожної пари різних рядків дорівнює 0, а скалярний добуток кожного рядка на себе дорівнює 1.

Використати масиви numpy та векторизувати програмний код.

20.11. Скласти програму, що перевіряє чи є задана квадратна матриця з цілих чисел магічним квадратом, тобто такою, в якій суми елементів в усіх рядках і стовпчиках однакові.

Використати масиви numpy та векторизувати програмний код.

20.12. Скласти програму, що перевіряє чи є задана квадратна матриця

а) верхньою трикутною

б) нижньою трикутною

Використати масиви numpy та векторизувати програмний код.

20.13. Задані координати n точок на площині $(x_1, y_1), \dots, (x_n, y_n)$. Знайти номери двох точок, відстань між якими найбільша (вважати, що така пара точок єдина), та саму відстань.

Використати масиви numpy. Точки розмістити у двовимірному масиві $2 \times n$. Побудувати тривимірний масив усіх можливих пар точок. Для побудови використати індексні масиви. Описати векторизовану функцію, яка обчислює відстань між 2 точками.

20.14. Задані координати n точок на площині $(x_1, y_1), \dots, (x_n, y_n)$. Знайти номери трьох точок, які утворюють трикутник найбільшого периметру, та сам периметр.

Використати масиви `numpy`. Точки розмістити у двовимірному масиві $2 \times n$. Побудувати тривимірний масив усіх можливих трійок точок. Для побудови використати індексні масиви. Описати векторизовану функцію, яка обчислює периметр трикутника, що утворений 3 точками.

20.15. Задані координати n точок на площині $(x_1, y_1), \dots, (x_n, y_n)$. Знайти кількість рівносторонніх трикутників, утворених цими точками.

Використати масиви `numpy`. Точки розмістити у двовимірному масиві $2 \times n$. Побудувати тривимірний масив усіх можливих трійок точок. Для побудови використати індексні масиви. Описати векторизовану функцію, яка перевіряє, чи є трикутник, утворений 3 точками, рівностороннім.

20.16. Скласти програму пошуку найменшого серед найбільших елементів рядків квадратної дійсної матриці порядку n , тобто $\|A\| = \min_{1 \leq i \leq n} \max_{1 \leq j \leq n} \{a_{ij}\}$.

Використати масиви `numpy` та векторизувати програмний код.

20.17. Виконати наближене обчислення π методом Монте-Карло. Для цього наближено обчислити площу півкола радіусом 1 з центром у початку координат. Використати масиви `numpy`. Зобразити на графіку півколо та охоплюючий прямокутник.

20.18. Вам пропонують зіграти у таку гру. Ви платите 1 одиницю грошей. Кидають 4 кості. Якщо сума не перевищить 9, Ви отримуєте 10 одиниць грошей. Чи будете Ви у виграві після багаторазового повторення гри?

Розв'язати задачу методом Монте-Карло з використанням масивів `numpy`.

Векторизувати програмний код.

Гра вважається чесною, якщо сума винагороди дорівнює витраченим грошам, за умови ймовірного виграву. Справедлива сума винагороди при вкладанні у кожен гру 1 одиниці грошей становить $1/p$, де p – ймовірність виграву.

20.19. Нехай ви граєте у гру під назвою «кrap». Гра полягає у наступному. Ви кидаєте 2 кості. Якщо сума кrapок на поверхнях костей буде 7 або 11 (випаде 7 або 11), - ви виграєте одразу. Якщо випаде 2, 3 або 12, - ви програєте одразу. Якщо ж випаде інше число (яке в подальшому називається «ваше число»), - ви продовжуєте кидати кості, поки не випаде 7 (у цьому випадку Ви програєте) або ваше число (у цьому випадку Ви виграєте). Яка ймовірність виграшу у цю гру?

Розв'язати задачу методом Монте-Карло з використанням масивів numpy.

Векторизувати програмний код.

20.20. Нехай в завдання 20.19 Ви зробили ставку на результат гри у 1 одиницю грошей. У разі програшу, ви програєте свою ставку. Якщо ж Ви виграєте, то виграш визначається наступним чином:

- якщо Ви виграєте одразу, викинувши 7 або 11, Ви отримуєте виграш у розмірі ставки;
- якщо Ваше число 6 або 8 і ви виграєте, Ви також отримуєте виграш у розмірі ставки;
- якщо ж Ваше число 4 або 10 і ви виграєте, Ви отримуєте виграш у розмірі подвійної ставки;
- нарешті, якщо Ваше число 5 або 9 і ви виграєте, Ви отримуєте виграш у розмірі $3/2$ ставки.

Скільки грошей в середньому Ви виграєте або програєте за одну гру?

Розв'язати задачу методом Монте-Карло з використанням масивів numpy.

Векторизувати програмний код.

20.21. Нехай гравець у казино грає у гру, що складається з окремих незалежних раундів. Ймовірність виграшу раунду дорівнює p . Якщо після якогось раунду капітал гравця становить N одиниць грошей, то такий гравець визнається переможцем та видаляється з казино. Яка ймовірність того, що гравець під час гри рано чи пізно втратить всі гроші, якщо він має початковий капітал K ?

Розв'язати задачу методом Монте-Карло з використанням масивів `numpy`. Для моделювання можна вважати, що ймовірність p кратна 0.1 , тобто $p \cdot 10$ – ціле число.

Векторизувати програмний код.

20.22. Ще одна задача Шевальє де Мера, яку він ставив Блезу Паскалю, - це справедливий розподіл грошей між гравцями, якщо гра переривається, не дібівши кінця. Нехай є 2 гравці з однаковим хистом до гри: А та В. Нехай вони грають у гру, у якій за кожен виграний раунд дається один бал. Той, хто набрав n балів, виграє гру та забирає банк. В якій пропорції чесно розділити гроші між гравцями, якщо гру перервано, коли А набрав a балів, а В – b балів.

Зрозуміло, що чесним розподіл буде тоді, коли гроші будуть розподілені пропорційно ймовірності виграшу кожного з гравців.

Розв'язати задачу методом Монте-Карло з використанням масивів `numpy`.

20.23. У капелюсі є 12 кульок: по 4 кульки червоного, синього та чорного кольору. За один раз витягають 3 кульки. Яка ймовірність того, що з них не менше 2 чорних?

Розв'язати задачу методом Монте-Карло з використанням масивів `numpy`. Векторизувати програмний код, наскільки можливо.

20.24. У капелюсі є $m \cdot k$ кульок: по k кульок m кольорів ($m > 1$). За один раз витягають d кульок ($1 < d \leq k$). Яка ймовірність того, що всі вони одного кольору?

Розв'язати задачу методом Монте-Карло з використанням масивів `numpy`. Векторизувати програмний код, наскільки можливо.

20.25. Описати клас `Decks`, який призначений для моделювання великої кількості випробувань з роздавання гральних карт. Одна колода карт складається максимум з 52 карт (по 13 карт 4 мастей). Гідність карт від 2 до 10, а також валет, дама, король, туз. Мاستі – піки, трефа, бубни, черви. У тій чи іншій грі може встановлюватись обмеження щодо мінімальної гідності карт (наприклад, починаючи з 7). При одному роздаванні карт колода тасується випадковим чином та m гравцям роздають

по n карт. Інші карти залишаються в колоді. Гравець, якому роздають карти, називається «рукою».

Зовнішнє представлення карти – це кортеж з двох полів, що є рядками (<гідність>, <масть>). Для використання масивів `numpy` кожен карту у внутрішньому представленні можна закодувати цілим числом: гідність – від 2 (2) до 14 (туз), масть – 1, 2, 3, 4 помножити на 100. Так, наприклад, 9 трефа буде мати код 209. Для отримання масті карти k достатньо виконати $k // 100$, а для отримання гідності достатньо виконати $k \% 100$.

Клас `Decks` повинен містити методи для перетворення карти з зовнішнього представлення у внутрішнє та навпаки, метод роздавання карт, метод «фіксованого» роздавання та, можливо, інші методи. Метод роздавання карт повинен повертати тривимірний масив з N випадковим чином розданих колод (m гравцям по n карт всього $m*n*N$) а також двовимірний масив – залишків N колод. Метод фіксованого роздавання повинен фіксувати карти на першій руці та випадковим чином роздавати їх на всі інші руки.

Застосувати клас `Decks` для розв'язання задачі: 4 гравцям роздають по 5 карт. Обчислити ймовірність того, що на будь-якій руці опиняться:

а) 4 карти однакової гідності.

б) 5 карт однієї масті, гідність яких після сортування монотонно зростає на 1.

Розв'язати задачу методом Монте-Карло з використанням масивів `numpy`. Векторизувати програмний код, наскільки можливо.

20.26. В умовах попередньої задачі, зафіксувати якісь 5 карт на 1 руці, виконати фіксоване роздавання на інші руки та обчислити ймовірність того, що на будь-якій руці, окрім першої, опиняться:

а) 4 карти однакової гідності.

б) 5 карт однієї масті, гідність яких після сортування монотонно зростає на 1.

Розв'язати задачу методом Монте-Карло з використанням масивів `numpy`. Векторизувати програмний код, наскільки можливо.

20.27. Застосувати клас Decks для моделювання розкладів при грі у преферанс. У цю гру грають колодою з 32 карт, починаючи з 7, роздають 3 гравцям по 10 карт, 2 карти залишаються у «прикупі». Старшинство карт – у порядку зростання гідності. На основі аналізу власних карт один з гравців може вибороти право визначати гру. Цей гравець бере прикуп, скидає 2 «зайві» карти та оголошує козирну масть, яка «б'є» інші масті, а також кількість взяток, які він зобов'язується взяти. При кожному ході кожен з гравців кладе 1 карту та розігрується 1 взятка, яку забирає старша карта або козирна карта (старша з козирних карт, якщо їх декілька). Кожен з гравців зобов'язаний класти карту тієї масті, з якої зроблено хід. Якщо цієї масті немає, - то козирну карту. Якщо козирної карти немає, то будь-яку карту. Право наступного ходу отримує гравець, який взяв останню взятку.

Для того, щоб при власному першому ході гарантовано взяти всі 10 взяток треба мати у масті, яка буде оголошена козирною, одну з таких комбінацій карт:

- туз, король, дама, валет;
- туз, король, дама та 2 будь-які менші карти;
- туз, король та 4 будь-які менші карти;
- туз та 6 будь-яких менших карт;
- всі вісім карт однієї масті.

Треба також мати всі старші карти в усіх інших наявних мастях (або комбінацію у ще одній масті туза, короля, дами та 2 будь-яких менших карт за умови не більше 5 карт у козирній масті).

Знайти ймовірність наявності на будь-якій руці розкладу, що дозволяє за умови власного першого ходу гарантовано взяти всі 10 взяток:

- а) без урахування прикупу;
- б) з урахуванням прикупу.

Розв'язати задачу методом Монте-Карло з використанням масивів питру. Векторизувати програмний код, наскільки можливо.

20.28. В умовах попередньої задачі визначити ймовірність розкладів при грі у «мізер». Якщо один з гравців зобов'язується зіграти мізер, це означає зобов'язання за будь-яких умов не взяти

жодної взятки. Для того, щоб гарантовано не взяти жодної взятки, за умови не власного першого ходу, треба у кожній наявній масті мати одну з таких комбінацій карт:

- комбінацію з однієї карти 7
- комбінацію з 2 карт 7, 8
- комбінацію з 2 карт 7, 9
- комбінацію з 3 карт: одну з комбінацій з 2 карт плюс 9 (для 7, 8) або 10 або валет
- комбінацію з 4 карт: одну з комбінацій з 3 карт плюс 10 або валет (якщо 10 або валет немає у комбінації з 3 карт) або дама або король
- комбінацію з 5 або більше карт: одну з комбінацій з 4 карт плюс будь-які старші карти

Знайти ймовірність наявності на будь-якій руці розкладу, що дозволяє за умови не власного першого ходу гарантовано не взяти жодної взятки:

- а) без урахування прикупу;
- б) з урахуванням прикупу.

Розв'язати задачу методом Монте-Карло з використанням масивів numpy. Векторизувати програмний код, наскільки можливо.

20.29. В умовах задачі 20.28 визначити ймовірність розкладів при фіксованому роздаванні. Зафіксувати на першій руці карти так, щоб найкраща потенційно козирна масть складалась з 4 карт а також 2 карти, скинуті після взяття прикупу. Виконати фіксоване роздавання. Знайти ймовірності того, що на другій та третій руці інші 4 карти цієї масті роздано у співвідношенні 4:0, 1:3, 2:2, 3:1, 0:4.

20.30. Клас Drunkard2D, що моделює випадковий шлях у двовимірному просторі, реалізований наступним чином:

```
class Drunkard2D:
    '''Клас, що реалізує випадковий шлях у двовимірному
    просторі ("хода п'яниці").

    '''
    def __init__(self, num_drunkards, init_pos = None,
is_limited = False, bounds = None):
```

```

        self._n_d = num_drunkards           #кількість
точок
        self._is_limited = is_limited      #чи обмежена
область
        self._bounds = bounds              #границі
області (прямокутник)

        self._pos = init_pos               #позиції
всіх точок
        if self._pos is None:
            #якщо позиції не задано, встановлюємо всі
у (0,0)
            self._pos = np.zeros(self._n_d * 2)
            self._pos.shape = (2, self._n_d)
            if self._is_limited:
                #якщо задано границі, встановлюємо всі
точки у середину області
                xmin, ymin, xmax, ymax = self._bounds
                x = (xmin + xmax) // 2
                y = (ymin + ymax) // 2
                self._pos += np.array([[x], [y]])

        self._dirs = np.array([[[-1, 0], [0, -1], [1,
0], [0, 1]]) #можливі рухи
        self._dirs = np.transpose(self._dirs)
#зручніше мати транспонований масив

        self.fig_count = 0 #номер рисунку

        plt.hold(False) #кожного разу буде малювати
нове зображення,
#a не доповнювати
попереднє

@property
def bounds(self):
    '''Властивість границі (читання).'''
    return self._bounds

@bounds.setter
def bounds(self, new_bounds):
    '''Властивість границі (встановлення).'''
    self._bounds = new_bounds

@property

```



```

def pos(self):
    '''Властивість позиції точок (тільки
читання).'''
    return self._pos

def _push_into_bounds(self):
    '''Повернути всі точки у межі області.'''
    xmin, ymin, xmax, ymax = self._bounds
    self.pos[0][self.pos[0] < xmin] = xmin + 1
    self.pos[0][self.pos[0] > xmax] = xmax - 1
    self.pos[1][self.pos[1] < ymin] = ymin + 1
    self.pos[1][self.pos[1] > ymax] = ymax - 1

def step(self):
    '''Зробити один крок у моделюванні.'''
    #масив індексів для подальшого формування
масиву приростів
    ids = np.random.random_integers(0, 3,
self._n_d)
    # print(ids)
    # print(self._dirs)

    #масив приростів чергового кроку
    dxy = self._dirs[:,ids]
    # print(dxy)
    self._pos += dxy
    if self._is_limited:
        self._push_into_bounds()

def msteps(self, m):
    '''Зробити m кроків у моделюванні.'''
    for i in range(m):
        self.step()

def plot(self):
    '''Побудувати графік стану моделі.'''
    #set axes
    if self._bounds is None:
        xmin = ymin = -100
        xmax = ymax = 100
    else:
        xmin, ymin, xmax, ymax = self._bounds
    plt.plot(self._pos[0], self._pos[1], 'ob')
    plt.axis([xmin, xmax, ymin, ymax])

def show(self):

```

```

'''Побудувати та показати графік стану
моделі.'''
self.plot()
plt.show()

def savefig(self, path):
'''Побудувати та зберегти графік стану моделі
у файлі.

path - шлях до файлу, включаючи фінальний
символ
поділу каталогів ('/' або '\\').
Файл має ім'я відповідно масці
tmpXXXXX.png, деXXXXX - номер рисунку
'''
self.fig_count += 1
fname = "tmp{:0>5}.png".
format(self.fig_count)
self.plot()
plt.savefig(path + fname)

```

Описати аналогічним чином клас DrunkardND, який моделює випадковий шлях у n-вимірному просторі та зробити аналог класу Drunkard2D його наступником. Для показу розташування частинок виконувати проєкцію n-вимірного простору на деяку площину.

Змодельовати процес розповсюдження молекул газу у замкненій області у n-вимірному просторі. Показати цей процес у вигляді відео (анімації).

20.31. В умовах завдання T20.30 (реалізувати клас DrunkardND) розв'язати наступну задачу. Молекули двох газів розташовані у сусідніх частинах прямокутної області, які поділяються стінкою. Потім стінку прибирають та молекули починають рухатись за правилами випадкового шляху.

Змодельовати рух молекул протягом визначеної кількості кроків. Показати цей процес у вигляді відео (анімації), зобразивши молекули різних газів різними кольорами.

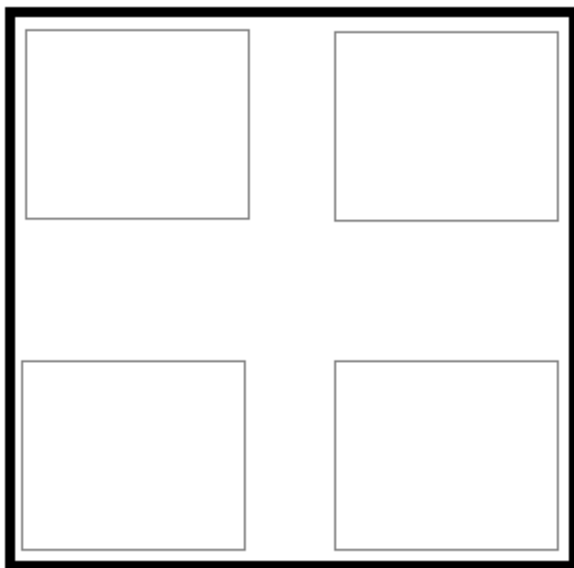
20.32. В умовах попереднього завдання 20.30 (реалізувати клас DrunkardND) розв'язати наступну задачу. Змодельовати рух множини частинок у одновимірному просторі, починаючи з

точки 0. Провести ряд випробувань для визначеної кількості кроків: 10, 100, 1000, ... Побудувати графіки залежності максимальної та середньої відстані, на яку віддаляються частинки, в залежності від кількості кроків. Використати звичайну та логарифмічну шкалу.

20.33. Описати функцію, яка робить циклічний k -зсув масиву NumPy направо. Тобто елемент з індексом 0 стає k -м, з індексом $1 - k + 1$ тощо.

Векторизувати програмний код

20.34. Дано квадратну матрицю A розміром $n \times n$. Перевірити, чи є у ній 4 однакові квадратні підматриці, які розташовано у кутах A , та мають розмір $k \times k$, де $k = 1, 2, \dots, n // 2$. Якщо є, то знайти підматриці максимального розміру



20.35. Перевірити, чи є матриця симетричною. Векторизувати програмний код.

20.36. Дано одновимірний масив numpy з цілих чисел. Вибрати з нього усі непарні числа.

Векторизувати програмний код. Використання циклів не передбачене.

20.37. Дано одновимірний масив numru з цілих чисел. Отримати новий масив, в якому усі непарні числа замінити на задане число m .

Векторизувати програмний код. Використання циклів не передбачене.

20.38. Дано двовимірний масив numru . Обміняти його стовпчики з індексами i та j .

Векторизувати програмний код. Використання циклів не передбачене.

20.39. Дано двовимірний масив numru . Обміняти його рядки з індексами i та j .

Векторизувати програмний код. Використання циклів не передбачене.

20.40. Дано двовимірний масив numru . Записати його стовпчики в оберненому порядку.

Векторизувати програмний код. Використання циклів не передбачене.

20.41. Дано двовимірний масив numru . Записати його рядки в оберненому порядку.

Векторизувати програмний код. Використання циклів не передбачене.

20.42. Дано двовимірний масив numru з цілих чисел. Вставити задане значення m у k випадкових позицій масиву.

Векторизувати програмний код. Використання циклів не передбачене.

20.43. Дано двовимірний масив numru з цілих чисел. Отримати двовимірний масив тих рядків першого масиву, які не містять нульових значень.

Векторизувати програмний код. Використання циклів не передбачене.

20.44. Дано двовимірний масив numru , що містить ненульові значення. Обчислити масив з відношень значень максимального та мінімального елементів рядків.

Векторизувати програмний код. Використання циклів не передбачене.

20.45. Скласти програму, яка вводить з клавіатури декілька значень v_0 та для кожного v_0 будує графік

$$y(t) = v_0 t - 0.5gt^2$$

(вважати $g = 9.81$). t належить інтервалу $[0, 2v_0/g]$, що означає, що для кожного v_0 буде власний масив точок графіку. Зобразити всі графіки у одному вікні. Застосувати різні стилі та кольори графіків, легенди графіків. На легенді зобразити відповідне значення v_0 .

20.46. Комп'ютер та гравець грають у таку гру:

Гра складається з m раундів кидання 2 костей. Перед кожним раундом комп'ютер та гравець намагаються вгадати суму n , яка випаде на 2 костях. Якщо після кидання сума на 2 костях дорівнює n , той, хто передбачив це, отримує n одиниць грошей. Якщо сума не дорівнює передбаченій, гравець або комп'ютер віддає 1 одиницю грошей.

Початковий бюджет комп'ютера та гравця – r одиниць грошей. В результаті гри програє той, хто має менше грошей або чий бюджет вичерпався (став рівним 0) по ходу гри.

Стратегія гри комп'ютера – на кожному раунді вибирати випадково суму зі значень від 2 до 12. Запропонувати стратегію гри гравця (вгадування суми), яка має бути краще стратегії гри комп'ютера. Провести велику кількість ігор, та перевірити (підтвердити) перевагу стратегії гравця.

Параметри гри (m , r) вибрати самостійно.

Векторизувати програмний код.

20.47. Описати функцію, яка переносить усі від'ємні елементи масиву `numru` до початку масиву, а інші – до кінця. Зберегти взаємний порядок розташування серед від'ємних та інших елементів. Векторизувати програмний код.

Тема 21 Регулярні вирази (re)

21.1. У текстовому файлі є дати, задані у форматі `dd.mm.yyyy` або у форматі `yyyy/mm/dd`. Також день та/або місяць може містити одну цифру, а не 2. Привести всі дати до єдиного формату `dd.mm.yyyy`.

Вказівка: використати регулярні вирази та функцію (метод) sub.

21.2. За допомогою регулярних виразів розбити текст у текстовому файлі на речення.

21.3. У текстовому файлі, що містить текст українською мовою, є дати (або сторіччя), записані римськими цифрами. Виділити всі такі числа та перевірити їх правильність. Число, записане римськими цифрами, використовує символи M (1000) D (500) C (100) L (50) X (10) V (5) I (1). При цьому, 900 записують як CM, 400 – CD, 90 – XC, 40 – XL, 9 – IX, 4 – IV. Наприклад, 1984 записується MCMLXXXIV.

Підказка: використати два шаблони регулярних виразів: один – для виділення числа, інший – для перевірки правильності.

21.4. У текстовому файлі містяться дати у форматі dd.mm.yyyy або підкреслення для запису дат вручну __.__.____ Знайти всі дати у тексті. Замість підкреслень вставити поточну дату. Зберегти оновлений текст.

21.5. У текстовому файлі містяться дати у форматі dd.mm.yyyy або yyyy-mm-dd або yyyy/mm/dd

Привести дати до єдиного формату за допомогою шаблону(iv), що містить(ять) іменовані підгрупи.

21.6. У текстовому файлі міститься переписка декількох осіб електронною поштою. Скласти список контактів (адрес електронної пошти) осіб, що фігурують у даній переписці. Використати регулярні вирази.

21.7. У текстовому файлі міститься текст українською мовою. «Стиснути» цей текст, видаливши у словах усі голосні літери. Якщо у слові тільки голосні літери або слово має довжину не більше 2 символів, - голосні не видаляти. Використати регулярні вирази.

21.8. У текстовому файлі містяться, крім іншої інформації, дійсні числа у форматі з фіксованою крапкою. При цьому, частина чисел не містить 0 перед крапкою, якщо число менше 1 за модулем (.253) або після крапки, якщо число ціле (5891.). Виділити всі дійсні числа, записані у файлі, та вставити 0 у тих числах, у яких його немає. Зберегти оновлений текст.

21.9. У текстовому файлі записано переписку у чаті, яка окрім слів містить «смайли». Виділити всі слова та всі смайли та порахувати «індекс емоційності» переписки, як відношення кількості смайлів до кількості слів.

21.10. В умовах задачі 21.9 розділити смайл за емоційністю на «просто емоційні» та «винятково емоційні». Просто емоційним смайлам присвоїти коефіцієнт 1, а винятково емоційним, - 2. Порахувати «індекс емоційності» переписки, враховуючи ці коефіцієнти.

21.11. Так звана «олбанська мова» - це інтернет-діалект російської мови з навмисним перекручуванням звичайних слів. У деяких словах окремі дзвінки приголосні замінюються глухими (аффар) або 'в' на 'фф', глухі, - дзвінками (йазыг), 'а' на 'о', 'о' на 'а', 'я' на 'йа', 'и' на 'ы', 'и' на 'е'. У текстовому файлі записано текст російською мовою. На основі цього тексту за допомогою регулярних виразів побудувати текст «олбанською мовою». Для побудови проводити від 1 до 2 замін у частині слів. Випробувати заміну 50%, 40%, 30%, 20% слів. Запропонувати додатково власні правила заміни.

21.12. За допомогою регулярних виразів проаналізувати синтаксичну правильність простих арифметичних виразів, що містять числа, знаки операцій, дужки. Наприклад, $2 - 57*(33 + 25/4)$.

21.13. У текстовому файлі зберігається текст лінійної програми у Python. У програмі використовується тільки команда присвоєння. У правій частині присвоєнь розташовані арифметичні вирази, що використовують арифметичні операції та містять константи та змінні. За допомогою регулярних виразів побудувати словник змінних програми (у формі <ім'я>:<значення>). Перевірити, чи є зміні, що використовуються раніше, ніж визначаються.

21.14. У текстовому файлі зберігається стаття. Треба скласти програму, яка перевіряє, чи має ця стаття позитивну чи негативну спрямованість. Про спрямованість статті свідчить використання слів: при позитивній спрямованості кількість позитивних слів суттєво перевищує кількість негативних. Використати словник, що складається з позитивних та негативних слів. Побудувати

регулярні вирази для знаходження різних словесних форм позитивних та негативних слів.

21.15. Ви працюєте у компанії, що надає послуги. Компанія має багато філій. Робітники на місцях зібрали дані про боржників компанії. Усі дані містять прізвище, ім'я та по-батькові, адресу боржника, його телефон та борг. Але порядок слідування інформації у різних філій – різний. Так само, розрізняється формат окремих частин інформації. Наприклад, адреса може починатися словом «адреса» або «адр.» у верхньому або нижньому регістрі. Так само, для багатоквартирних будинків може бути вказано «кв.» та номер квартири, а для приватних будинків номер квартири не вказаний. Замість прізвища, ім'я та по-батькові може бути надане прізвище та ініціали. Усю інформацію зібрано у текстовий файл. Інформація про кожного боржника займає 1 рядок файлу.

Всього боржників порядку 10 000. Ваше керівництво доручило Вам до наступного ранку підготувати до відправки листи всім боржникам такого змісту:

«<Адреса>

Шановна(ий) _____ <П.І.Б> _____

Сума Вашого боргу за послуги складає <Борг>.

Просимо сплатити борг протягом місяця. У іншому випадку, надання послуг буде припинено.»

Відомо, що адреса починається з одного зі слів вище, телефон починається з слів «телефон» або «тел.» (літери можуть бути у верхньому або нижньому регістрі. Частини інформації розділяються одним або декількома пропусками, сума боргу йде останньою у рядку.

Використайте регулярні вирази для розв'язання задачі та складіть програму для формування листів та запису їх у текстовий файл.

21.16. В умовах завдання 21.15 Вам доручили також терміново надати список телефонів 100 найбільших боржників для їх подальшого інформування телефоном.

Список повинен мати формат:

<П.І.Б> <Телефон> <Борг>

Використайте регулярні вирази для розв'язання задачі та складіть програму для формування списку.

21.17. Ви працюєте у департаменті інформаційної безпеки компанії. Відомо, що працівникам компанії заборонено на роботі користуватись рядом сайтів (є список адрес заборонених сайтів). На запит вашої компанії інтернет-провайдер надав інформацію про відвідування сайтів співробітниками за декілька місяців. Інформація – у вигляді великого текстового файлу, у якому, зокрема, є трійки:

<адреса комп'ютера> <відвіданий сайт> <дата та час>

<адреса комп'ютера> - рядок у форматі XXX.XXX.XXX.XXX, де X – цифри (між крапками може бути й менше 3 цифр.

<відвіданий сайт> - рядок у форматі http:// <адреса сайту>

<дата та час> - рядок у форматі dd.mm.yyyy hh:mm:ss

У Вас є список співробітників разом з адресами їх комп'ютерів.

Вам потрібно підготувати звіт щодо відвідування співробітниками заборонених сайтів, впорядкувавши його за кількістю відвідувань.

Використайте регулярні вирази для розв'язання задачі та складіть програму для формування списку.

21.18. У текстовому файлі зберігається авторська стаття. Треба скласти програму, яка аналізує цю статтю та виділяє найбільш часто використовувані слова, а також словосполучення з 2 та 3 послідовних слів.

Порівняти результат цієї статті з результатами для інших статей цього ж автора, а також для статей інших авторів.

Використати регулярні вирази.

21.19. У текстовому файлі записано список осіб. У 1 рядку – 1 особа. Дані про особу мають такий вигляд:

- Прізвище ім'я по-батькові
- Прізвище ініціали
- Ініціали прізвище

Побудувати список у форматі:

- Прізвище ініціали

для усіх осіб

21.20. У текстовому файлі є роки, записані римськими цифрами. Показати список років у порядку зростання. Вважати, що числа записані правильно.

Для розв'язання задачі скласти функцію, що перетворює рядок – запис числа римськими цифрами – у відповідне натуральне число.

Число, записане римськими цифрами, використовує символи М (1000) D (500) С (100) L (50) X (10) V (5) I (1). При цьому, 900 записують як CM, 400 – CD, 90 – XC, 40 – XL, 9 – IX, 4 – IV. Наприклад, 1984 записується MCMLXXXIV.

21.21. Виділити з текстового файлу усі IP-адреси та видалити з цих адрес лідируючі нулі. IP-адреса має формат:

XXX.XXX.XXX.XXX

де X – цифри. У кожній групі може бути й менше 3 цифр. Використати регулярні вирази.

21.22. Видалити з текстового файлу усі слова довжиною менше n символів. Слова можуть розділятися пропусками, розділовими знаками та містити тільки літери. Використати регулярні вирази.

21.23. Запропонувати шаблон, якому відповідають усі рядки з першого стовпчика та не відповідає жоден рядок з другого стовпчика

pit	pt
spot	Pot
spate	peat
slap two	part
respice	

21.24. Порахувати кількість слів у текстовому файлі. Використати регулярні вирази. Слова можуть поділятися довільними розділовими знаками.

21.25. Дано текстовий файл. Виділити з нього усі рядки, які містять слова з заданого списку (наприклад, error, warning, critical). Програма повинна мати такі налаштування:

- Регістр: підтримувати/ігнорувати
- Частина слова/слово цілком

Тема 22 Використання операційних систем (os, io)

22.1. Скласти програму порівняння двох каталогів. Програма повинна показувати різницю у каталогах, тобто файли, що присутні в одному каталозі та відсутні у іншому. Якщо файл присутній у одному та іншому каталозі, він не повинен бути показаний. Результат роботи програми слід спрямувати у текстовий файл.

22.2. Скласти програму порівняння файлів з однаковим ім'ям у двох каталогах. Для таких пар файлів треба показати різницю у часі створення файлу, тобто, який з двох файлів цієї пари був створений пізніше. Якщо файли були створені одночасно, то нічого показувати не потрібно. Результат роботи програми слід спрямувати у текстовий файл.

22.3. Скласти програму порівняння файлів з однаковим ім'ям у двох каталогах. Для таких пар файлів треба показати різницю у розмірі файлів. Якщо файли мають однаковий розмір, то нічого показувати не потрібно. Результат роботи програми слід спрямувати у текстовий файл.

22.4. Деяка існуюча програма записує у системний журнал. Системні журнали містяться у заданому каталозі та створюються по декілька файлів на кожну дату. Відповідно, дата та час створення є частиною імені файлу.

Треба скласти програму, яка архівує системні журнали у заданому каталозі за попередні дати у форматі tar.gz.

22.5. Деяка існуюча програма записує у системний журнал. Системні журнали містяться у заданому каталозі та створюються по декілька файлів на кожну дату. Відповідно, дата та час створення є частиною імені файлу. Вважається, що системні журнали повинні зберігатися протягом певного терміну, наприклад, одного місяця, після чого можуть бути знищені.

Треба скласти програму, яка знищує усі системні журнали у заданому каталозі, термін зберігання яких вичерпано. Системні

журнали – це файли, які мають розширення імені .log або архіви .tar.gz.

22.6. Деяка існуюча програма записує у системний журнал. Системні журнали містяться у заданому каталозі та створюються по декілька файлів на кожен дату. Відповідно, дата та час створення є частиною імені файлу.

Описати функцію, яка отримує в якості аргументу постійну частину імені файлу системного журналу, сам об'єкт – відкритий файл системного журналу.

Далі ця функція повинна перевірити, чи перевищує розмір файлу системного журналу задану величину у байтах.

Якщо так, то закрити старий файл, створити новий файл системного журналу, додавши поточну дату та час до імені файлу та повернути його ім'я та відкритий об'єкт нового файлу.

Якщо ні, то повернути отримані аргументи без змін.

Перевірити роботу цієї функції для програми, що пише у системний журнал рядки прочитаного текстового файлу.

22.7. Скласти програму, яка знаходить у каталозі та його підкаталогах «найстарший» файл, тобто файл, створений раніше за інших.

22.8. Скласти програму, яка зберігає склад усіх підкаталогів даного каталогу. Для усіх файлів та підкаталогів треба зберігати у текстовому файлі ім'я, дату та час їх зміни. Передбачити також порівняння інформації про склад каталогу за різні дати та час. Якщо з'явилися нові файли або каталоги, знищено раніше існуючі або змінилась дата/час зміни існуючих файлів або каталогів, - інформація про це повинна бути показана (збережена у окремому файлі).

Використати цю програму для перевірки того, на які каталоги впливає інсталяція/деінсталяція деякої програми.

22.9. Скласти програму, яка здійснює пошук у заданому каталозі та його підкаталогах усіх текстових файлів, імена яких відповідають заданій масці, та у яких міститься рядок, що відповідає заданому регулярному виразу.

22.10. Скласти програму, яка здійснює заміну новим рядком рядка, що відповідає заданому регулярному виразу, у знайдених

у заданому каталозі та його підкаталогах усіх текстових файлах, імена яких відповідають заданій масці.

22.11. При відправленні великого обсягу даних електронною поштою у вигляді вкладення часто виникає ситуація, коли поштовий сервер не пропускає дане вкладення через перевищення допустимого розміру. Скласти програму, яка робить багатотомний архів заданого каталогу. Програма повинна спочатку зробити архівний файл. Потім перевірити, чи перевищує розмір архіву заданий максимальний розмір. Якщо так, то розбити архів на частини - окремі файли заданого максимального розміру (розмір останньої частини може бути меншим). Кожна частина повинна мати ім'я, яке починається з імені архіву, за яким стоїть порядковий номер частини: 0001, 0002 тощо.

22.12. Скласти програму, яка зливає багатотомний архів (див. завдання 22.11) з файлів-частин у один архівний файл та розкрити цей архів у заданий каталог.

22.13. У деякому каталозі у окремих текстових файлах зберігаються замітки. Скласти програму, яка з'єднає усі текстові файли з заданого каталогу у один текстовий файл. При з'єднанні слід вибирати файли у порядку часу їх створення (спочатку – файли які створені раніше).

22.14. Скласти програму, яка обчислює статистику по файлах з різними розширеннями у заданому каталозі. Для кожного розширення програма повинна показати кількість та загальний розмір у байтах файлів з цим розширенням. Також здійснити перенаправлення результатів роботи програми у текстовий файл.

22.15. Скласти програму, яка порівнює усі пари підкаталогів заданого каталогу на предмет наявності в них файлів з однаковими іменами. Якщо відсоток таких файлів для якоїсь пари каталогів перевищує заданий поріг, - показати ці каталоги та файли. Також здійснити перенаправлення результатів роботи програми у текстовий файл.

22.16. Скласти програму, яка запускає по графіку програму з завдання 22.8 та видає попереджувальне повідомлення, якщо виникли зміни у заданих каталогах.

22.17. Скласти програму, яка у дереві каталогів, що починається з заданого каталогу, буде видаляти файли, які є дублями існуючих файлів. Тобто, з декількох копій 1 файлу залишати 1 копію у каталозі вищого рівня. Вважати, що файли є копіями можна, якщо вони мають однакове ім'я та розмір.

22.18. Скласти програму, яка зберігає інформацію про файли з заданим розширенням, які містяться у дереві каталогів. Для кожного розширення інформацію зберігати у текстовому файлі у вигляді повних шляхів до файлів з відповідним розширенням. Ці текстові файли мають називатись так, як і розширення з розширенням `txt`. Наприклад, `exe.txt` для файлів з розширенням `.exe`.

22.19. У дереві каталогів, що починається з заданого каталогу, містяться тільки текстові файли. Знайти (якщо є) два таких файли, які мають однаковий розмір та розрізняються не більше, ніж у n позиціях, де n – задане натуральне число.

Тема 23 Робота з даними у офісних документах

23.1. У деякому каталозі містяться документи MS Word. У кожному документі є заголовки 1 та 2 рівня (стили `Heading 1` та `Heading 2`) а також звичайний текст (стиль `Normal`). Один з документів у каталозі вважається базовим, тобто таким, у якому форматування має еталонний вигляд.

Скласти програму, яка робить форматування всіх документів MS Word у каталозі ідентичним форматуванню базового документу. Під форматуванням мається на увазі форматування параграфу та тексту.

23.2. У документі Word є дати, задані у форматі `dd.mm.yuuu` або у форматі `uuu/mm/dd`. Також день та/або місяць може містити одну цифру, а не 2. Привести всі дати до єдиного формату `dd.mm.yuuu`.

23.3. У документі Word, що містить текст українською мовою, є дати (або сторіччя), записані римськими цифрами. Виділити всі такі числа та перевірити їх правильність. Число, записане римськими цифрами, використовує символи M (1000) D (500) C (100) L (50) X (10) V (5) I (1). При цьому, 900 записують як CM, 400 – CD, 90 – XC, 40 – XL, 9 – IX, 4 – IV. Наприклад, 1984 записується MCMLXXXIV.

23.4. У документі Word містяться дати у форматі dd.mm.yyyy або підкреслення для запису дат вручну Знайти всі дати у тексті. Замість підкреслень вставити поточну дату. Зберегти оновлений документ.

23.5. У документі Word містяться дати у форматі dd.mm.yyyy або yyyy-mm-dd або yyyy/mm/dd

Привести дати до єдиного формату.

23.6. У документі Word містяться, крім іншої інформації, дійсні числа у форматі з фіксованою крапкою. При цьому, частина чисел не містить 0 перед крапкою, якщо число менше 1 за модулем (.253) або після крапки, якщо число ціле (5891.). Виділити всі дійсні числа, записані у файлі, та вставити 0 у тих числах, у яких його немає. Зберегти оновлений документ.

23.7. Скласти програму, яка здійснює пошук у заданому каталозі та його підкаталогах усіх документів MS Word, у яких міститься рядок, що відповідає заданому регулярному виразу. Для кожного файлу, у якому знайдено такий текст, повернути номер параграфу з відповідним текстом та текст параграфу.

23.8. Скласти програму, яка здійснює заміну новим рядком рядка, що відповідає заданому регулярному виразу, у знайдених у заданому каталозі та його підкаталогах усіх документах MS Word.

23.9. Скласти програму, яка здійснює пошук у заданому каталозі та його підкаталогах усіх робочих книг MS Excel, у яких міститься рядок, що відповідає заданому регулярному виразу. Для кожного файлу, у якому знайдено такий текст, повернути назву робочого аркуша та адресу клітинки з відповідним текстом.

23.10. У аркуші робочої книги MS Excel містяться дані про проекти та осіб, задіяних у цих проектах у формі:

Project	Person
---------	--------

Скласти програму, яка у окремому робочому аркуші буде список зв'язків між особами у формі:

Person1	Person2	Project

23.11. У аркуші робочої книги MS Excel містяться дані про проекти та осіб, задіяних у цих проектах у формі:

Project	Person

Скласти програму, яка у окремому робочому аркуші буде список зв'язків між особами у формі:

Person1	Person2	Projects	Weight

де Проекти (Projects) – це всі проекти через підкреслення, на яких спільно працювали дві особи, а вага (Weight) зв'язку, - кількість спільних проектів двох осіб.

23.12. У робочий книзі MS Excel містяться дані про авіарейси. На аркуші «Авіакомпанії» - назва та код авіакомпанії, наприклад

Id	Name
PS	Ukraine International Airlines
AF	Air France

--	--

На аркуші «Аеропорти» - код та назва аеропорту, а також назва міста, наприклад:

Id	Airport	City
KBP	Бориспіль	Київ
IEV	Київ	Київ
CDG	Шарль-де-Голль	Париж

На аркуші «Рейси» - інформація про рейси, наприклад:

from_Id	to_Id	Flight	Days	Depart	Arrive	Class	Cost
KBP	CDG	AF2268	1030060	10:35	13:50	E	8350
IEV	CDG	PS765	0204060	10:35	13:50	E	7500

де Days – номери днів тижня, коли здійснюється рейс (0 означає, що у цей день рейс не здійснюється)

Скласти програму, яка знаходить:

- найдешевший переліт з одного міста до другого
- найшвидший переліт з одного міста до другого

23.13. В умовах завдання 23.12 побудувати аркуш у робочій книзі MS Excel, який містить розклад вильоту рейсів з заданого аеропорту на задану дату.

23.14. Велика компанія веде список зареєстрованих постачальників для здійснення закупівель. Кожен постачальник має номенклатуру продукції, яку він постачає, та рейтинг у балах на підставі попередніх закупівель.

Дані зберігаються у робочій книзі MS Excel.

На аркуші «Постачальники» зберігається код постачальника, назва, рейтинг, наприклад:

Id	Name	Rating	Adress
S01	Доміно	4	domino@com.ua
S02	Кондор	6	condor@com.ua

На аркуші «Продукція» зберігається код товару та назва, наприклад:

Id	Name
P01	Олівець
P02	Ручка кулькова

На аркуші «Ціна» зберігається код товару, код постачальника, ціна та термін поставки у днях, наприклад:

S_id	P_id	Price	Term
S01	P01	2,5	5
S02	P01	2,4	6

Нехай, у тендері на закупівлі для відбору найкращого постачальника вибір по кожному товару здійснюється за найбільшим значенням, що отримується з формули:

$$S = a_1 * P / P_{\max} + a_2 * R / R_{\max},$$

де a_1 , a_2 - коефіцієнти, що в сумі дають 1, P – ціна товару у даного постачальника, P_{\max} - максимальна ціна даного товару, R – рейтинг даного постачальника, R_{\max} - максимальний рейтинг усіх постачальників.

Скласти програму, яка для заданого переліку товарів відбирає найкращого постачальника для кожного товару при заданих коефіцієнтах a_1 , a_2 .

23.15. Скласти програму, яка в умовах завдання 23.14 для заданого переліку товарів відбирає постачальника кожного товару, який здійснить найшвидшу поставку.

23.16. В умовах завдання 23.14 вважати, що ціна на аркуші «Ціна» є індикативною та може змінюватись.

Скласти програму, яка для кожного товару з заданого переліку товарів відбирає не більше трьох кращих, з точки зору рейтингу, постачальників та формує листи-запрошення на участь у тендері та уточнення ціни.

Лист-запрошення повинен формуватися у MS Word та мати орієнтовно такий зміст:

«Пропонуємо взяти участь у тендері та уточнити ціну на продукцію з переліку нижче:

1. <Продукт 1>
2. <Продукт 2>
3. ...»

Для кожного з відібраних постачальників у лист-запрошення повинна входити тільки продукція, яку постачає цей постачальник, та яка фігурує у тендері.

23.17. У робочій книзі MS Excel зберігається інформація про замовлення товарів у продавця.

На аркуші «Покупці» міститься інформація про покупців, наприклад:

Id	Name	Address
C01	Доміно	domino@com.ua
C02	Кондор	condor@com.ua

На аркуші «Товари» міститься код та назва товару а також одиниця виміру та ціна, наприклад:

id	Name	Unit	Price
P01	Олівець	шт.	2,5
P02	Ручка кулькова	шт.	2,4

На аркуші «Рахунки» міститься код, номер та дата рахунку а також код покупця, наприклад:

id	No	Date	Client
I01	253	18.07.2016	C01
I02	255	19.07.2016	C02

На аркуші «Пункти» міститься інформація про пункти рахунку. А саме, код рахунку код товару, кількість, наприклад:

I_id	P_id	Quantity
I01	P01	200
I01	P02	150

Скласти програму, яка готує документ MS Word, що містить рахунок із заданим номером. Документ повинен мати орієнтовно такий зміст:

«

Рахунок № _____

Дата __.__.____

Покупець:

№	Назва	Од. виміру	Кількість	Ціна	Сума

Всього _____

»

23.18. В умовах завдання 23.17 сформувати рахунок вказаного змісту у робочому аркуші нової книги MS Excel.

23.19. В умовах завдання 23.17 побудувати список покупців, які за вказаний період купили товар у порядку зменшення загальної суми закупівель.

23.20. В умовах завдання 23.17 побудувати список товарів, які за вказаний період користувалися попитом у порядку зменшення загальної суми закупівель.

23.21. Компанія виконує великий проект, у рамках якого на багатьох майданчиках компанії-замовника паралельно виконуються схожі роботи. Необхідно автоматизувати друк актів виконаних робіт.

Інформація про виконання робіт зберігається у робочій книзі MS Excel.

На аркуші «Майданчики» міститься інформація про код, назву, адресу майданчика та прізвища відповідальних осіб від замовника та виконавця, наприклад:

Id	Name	Address	Responsible	Manager
----	------	---------	-------------	---------

S01	Рубежанська філія ТТС	Рубіжне, вул. Шевченка, 22	Іваненко І.І.	Костенко К.К.
S02	Перечинська філія ТТС	Перечин, вул. Шкільна, 14	Петренко П.П.	Костенко К.К.

На аркуші «Роботи» міститься інформація про код роботи та назву роботи, наприклад:

Id	Name
W01	Прокладання кабелю 25 м
W02	Тестування обладнання

На аркуші «Акти» міститься інформація про код акту, номер акту, дату акту, суму за актом, код майданчику, наприклад:

Id	No	Date	Sum	S_id
A01	34	18.07.2016	17580	S01
A02	75	19.07.2016	23250	S02

На аркуші «Пункти» міститься інформація про код роботи та код акту, наприклад:

W_id	A_id
W01	A01
W02	A01

Використовуючи інформацію з робочої книги, отримати документ MS Word, який містить усі акти виконаних робіт. Окремий акт повинен мати орієнтовно такий зміст:

«
 Акт № _____ Дата _____.____.

Даний Акт засвідчує, що Виконавцем на майданчику _____ за адресою _____ буди виконані такі роботи:

№	Назва роботи

Сума виконаних робіт складає _____ грн.

Від Замовника

Від Виконавця

»

Підказка: для розв'язання можна використати програму для злиття даних, модифікувавши її для заповнення таблиць робіт.

23.22. Заданий аркуш робочої книги MS Excel містить таблицю з даними. У першому рядку записано заголовки стовпчиків. Починаючи з другого рядка, - самі дані. Описати клас, який є класом-ітератором та на кожному кроці повертає словник з ключами – заголовками стовпчиків та значеннями – даними з клітинок поточного рядка таблиці. Клас також повинен мати метод, що повертає список заголовків стовпчиків.

23.23. Скласти програму, яка у файлі MS Word замінює кольором с2 всі входження тексту з кольором с1.

23.24. Переставити параграфи у документі MS Word так, щоб вони були впорядковані за неспаданням кількості слів.

23.25. У файлі MS Word є декілька таблиць. Відомо, що у деяких з цих таблиць у останньому стовпчику, починаючи з 2 рядка містяться грошові суми у вигляді чисел (по 1 числу у клітинці таблиці). Порахувати загальні суми по всіх рядках у тих таблицях, у яких вказані числа у останньому стовпчику.

Показати:

Номер таблиці у документі

Тексти першого рядка таблиці

Загальну суму

23.26. У робочій книзі MS Excel на вказаному аркуші у 2 стовпчиках, що йдуть підряд, є дані табулювання функції $f(x)$. У 1 рядку вказано «x» та ім'я функції. Починаючи з 2 рядка наведено значення x та $f(x)$.

23.27. Скласти програму, яка записує ці значення у масиви numpy та буде графік функції за допомогою matplotlib.

23.28. На єдиному робочому аркуші робочої книги MS Excel містяться елементи квадратної матриці $n \times n$. Перевірити, чи є ця матриця верхньою трикутною.

Переставити параграфи у документі MS Word так, щоб вони були впорядковані за неспаданням довжини тексту параграфу.

Тема 24 Робота з графікою та графічні інтерфейси (tkinter)

.....

24.1. Скласти програму з графічним інтерфейсом для введення x , ε та обчислення суми усіх доданків, які за абсолютною величиною не менші ε :

$$y = \frac{1}{(1+x)^2} = 1 - 2*x + 3*x^2 - \dots, \quad (|x| < 1);$$

Проконтролювати виконання умови $|x| < 1$. Показати результат обчислень.

24.2. Скласти програму з графічним інтерфейсом для введення рядка та перевірки, чи є цей рядок паліндромом. Показати результат перевірки.

24.3. Скласти програму з графічним інтерфейсом для введення рядка та виведення всіх різних слів цього рядка. Слова виводити у віджет список.

24.4. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Задана непорожня послідовність ненульових цілих чисел, за якою йде 0. Визначити кількість змін знаку в цій послідовності. Наприклад, у послідовності 1, -34, 8, 14, -5, 0 знак змінюється три рази.

Для введення елементів послідовності використати одне поле введення. Кнопка «Обробити» повинна ініціювати обробку введеного елементу послідовності та очищення поля введення.

Показати результат обчислень.

24.5. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Виключити з заданого рядка групи символів, які знаходяться між '(' та ')'. Самі дужки теж мають бути виключені. Вважається, що дужки розставлено правильно (парами) та всередині кожної пари дужок немає інших дужок.

Ввести рядок та показати результат обчислень.

24.6. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Знайти у даному рядку символ та довжину найдовшої послідовності однакових символів, що йдуть підряд.

Ввести рядок. Показати рядок, виділивши знайдену найдовшу послідовність іншим кольором фону та тексту.

24.7. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Вивести на друк усі цифри, які входять у заданий рядок, та окремо - решту символів, зберігаючи при цьому взаємне розташування символів у кожній з цих двох груп.

Ввести рядок та показати 2 рядки-результати.

24.8. Описати клас для введення вектору з n компонент у графічному режимі. Можна вважати, що поля введення для всіх компонент вміщуються на екрані. Перевіряти, чи всі поля заповнено. З використанням цього класу побудувати графічний інтерфейс для розв'язання задачі: ввести вектор з n дійсних компонент та знайти кількість компонент, що належать відрізку $[a, b]$. Вектор вводити у окремому діалоговому вікні. У головному вікні вводити n , a , b та показати компоненти введеного вектору у віджеті список а також результат обчислень.

24.9. Використати клас для введення вектору з попереднього завдання для розв'язання задачі: ввести два вектори з n дійсних компонент та обчислити їх скалярний добуток. Вектори вводити у окремому діалоговому вікні. У головному вікні вводити n та показати компоненти введених векторів у віджетах список а також результат обчислень.

24.10. Описати клас для введення матриці $n \times m$ у графічному режимі. Можна вважати, що поля введення для всіх елементів вміщуються на екрані. Перевіряти, чи всі поля заповнено. З використанням цього класу розв'язати задачу: ввести 2 матриці та обчислити їх добуток. Матриці вводити у окремому вікні. У

головному вікні ввести розміри матриць, показати у віджетах список введених матриць та матрицю, що є результатом (у одному рядку списку показувати один рядок матриці).

24.11. Описати клас для введення коефіцієнтів поліному степні n . Можна вважати, що поля введення для всіх коефіцієнтів вміщуються на екрані. Поля введення для нульових коефіцієнтів можна не заповнювати. З використанням цього класу розв'язати задачу: ввести 2 поліноми P_1 , P_2 та рядок, який містить вираз, що залежить від 2 поліномів. Наприклад,

$$P_1 + P_2 * P_1 - P_2$$

Обчислити поліном, який буде значенням цього виразу.

Поліноми вводити у окремому вікні. У головному вікні вводити степені поліномів та рядок виразу. Показувати у віджетах список введених поліномів та поліном-результат.

24.12. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Дано файл, який містить відомості про іграшки: вказується назва іграшки (наприклад: м'яч, лялька, конструктор і т.д.), її вартість в гривнях та вікові границі дітей, для яких іграшка призначається (наприклад, для дітей від двох до п'яти років). Підібрати усі іграшки за заданим віком дитини та/або обмеженням вартості.

Вводити відомості про іграшки треба у окремому вікні. У головному вікні вводити обмеження та показувати відібрані іграшки у віджеті список.

24.13. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Дано файл, який містить відомості про книги. Відомості про кожну книгу - це прізвище автора, назва та рік видання. Підібрати усі книги за заданою назвою та/або автором та/або періодом видання.

Вводити відомості про книги треба у окремому вікні. У головному вікні вводити обмеження та показувати відібрані книги у віджеті список.

24.14. Скласти програму з графічним інтерфейсом для розв'язання задачі.

У двох файлах містяться результати футбольного турніру. У першому файлі записано назви команд. У другому файлі – результати матчів у вигляді:

команда1 команда2 m1 m2

де, *команда1, команда2* – номери першої та другої команди;
m1, m2 – кількість м'ячів, забитих відповідно першою та другою командою.

За перемогу нараховується 3 очки, за нічию – 1, за поразку – 0.

З двох команд, які мають однакову кількість очок, першою вважається та, яка

- 1) має кращу різницю забитих та пропущених м'ячів;
- 2) при однаковій різниці має більше забитих м'ячів;
- 3) при всіх однакових попередніх показниках визначається жеребкуванням (використати для жеребкування генератор випадкових чисел).

Показати поточну таблицю турніру у вигляді:

місце, команда, ігор, вигравив, нічийх, поразок, м'ячів забито, м'ячів пропущено, очок

Графічний інтерфейс повинен надавати можливість введення результату матчу у окремому вікні. У головному вікні повинна відображатись поточна таблиця турніру.

24.15. Скласти програму з графічним інтерфейсом для конвертування валют. Поточні курси валют містяться у файлі Excel у форматі: <код валюти 1> <код валюти 2> <курс>. Код валюти – це рядок з 3 символів, наприклад UAH, USD, EUR тощо.

Currency1	Currency2	Rate
UAH	USD	24,81
UAH	EUR	27,75
...

Програма повинна забезпечити вибір двох валют зі списків кодів валют, введення суми у першій валюті та показ суми у другій валюті.

24.16. Скласти програму з графічним інтерфейсом для нагадування про подію. Програма повинна забезпечити введення дати та часу нагадування та опису події, сховати вікно та відкрити вікно нагадування у зазначений час. Забезпечити можливість збереження інформації про декілька подій та нагадування про кожен з них.

24.17. Скласти програму з графічним інтерфейсом для телефонного довідника. Дані про знайомих та їх телефони зберігаються у файлі. Програма повинна забезпечити додавання інформації про знайомого, редагування інформації про знайомого та пошук телефону за прізвищем.

24.18. Скласти програму з графічним інтерфейсом для введення таблицю та обчислення заробітної платні співробітників. Клас Person описано наступним чином:

```
class Person:                                     #Клас Особа
    def __init__(self):
        self.name = None                          #прізвище
        self.byear = None                         #рік народження

    def input(self):                               #ввести особу
        self.name = input('Прізвище: ')
        self.byear = input('Рік народження: ')

    def print(self):                               #вивести особу
        print(self.name, self.byear, end = ' ')
```

Описати клас Співробітник на базі класу Person.

У цьому класі повинно бути як мінімум два додаткових поля: «табельний номер» та «заробітна платня», а також методи введення та виведення інформації про співробітника та розрахунку заробітної платні за кількістю відпрацьованих годин та загальною кількістю годин на місяць.

Використати цей клас для розрахунку заробітної платні за місяць, якщо співробітник працює на умовах погодинної оплати. Для розрахунку вводити відпрацьовані години по днях місяця (табелі). Вважається, що відома кількість годин, яку повинен відпрацювати співробітник кожного дня, для отримання 100% платні.

Передбачити дії: введення таблицю для співробітника, введення таблиць всіх співробітників, розрахунку заробітної платні всіх співробітників.

Табель зберігати у файлі та читати з файлу.

Вводити таблиць для співробітника у окремому вікні. У головному вікні показувати список співробітників та список співробітників з їх заробітною платнею.

24.19. Скласти програму з графічним інтерфейсом для розрахунку плати за проживання.

Описати клас Гість на базі класу Person (див. завдання 24.18) та клас ТипНомера з полями «назва типу», «плата за добу».

У класі Гість повинно бути як мінімум два додаткових поля: «номер» та «кількість днів проживання», а також методи введення та виведення інформації про гостя та розрахунку плати за проживання.

Використати цей клас для реєстрації гостя у номері та розрахунку плати за проживання усіх гостей.

Номери та інформацію про гостей зберігати у файлах та читати з файлів.

Графічний інтерфейс повинен передбачати реєстрацію гостя у окремому вікні. При реєстрації вибрати номер зі списку номерів та вказати кількість днів проживання. У головному вікні показувати список гостей та плату за проживання.

24.20. Скласти програму з графічним інтерфейсом для розрахунку плати за проїзд.

Описати клас Пасажир на базі класу Person (див. завдання 24.18).

У класі Пасажир повинно бути як мінімум два додаткових поля: «місто відправлення» та «місто прибуття», а також методи введення та виведення інформації про пасажера та розрахунку плати за білет.

Використати цей клас для розрахунку плати за білети усіх пасажирів. Вважати що маршрути зберігаються у списку кортежів (місто1, місто2, відстань), а також те, що плата за білет пропорційна відстані та відома плата за 1 км відстані. Маршрути зберігаються у файлі.

Надати графічний інтерфейс для введення нового маршруту у окремому вікні а також для додавання пасажира у новому вікні. У головному вікні вивести список маршрутів, дати можливість вибрати маршрут, пасажира та обчислити та показати вартість проїзду пасажира. Зберегти дані пасажирів у файлі.

24.21. Скласти програму з графічним інтерфейсом для розрахунку плати за перевезення.

Описати клас Водій на базі класу Person (див. завдання 24.18).

У класі Водій повинно бути як мінімум два додаткових поля: «плата за тоно-кілометр» та «вантажопідйомність автомобіля», а також методи введення та виведення інформації про водія та розрахунку плати за перевезення вантажу на задану відстань.

Маршрутний лист водія за певну дату містить кількість кілометрів перевезень вантажу.

Використати цей клас для розрахунку плати всіх водіїв за період. Вважати що маршрутні листи водіїв зберігаються у списку кортежів (водій, дата, відстань), а також те, що плата за перевезення пропорційна добутку відстані на вагу вантажу. Дані водіїв зберігаються у файлі.

Графічний інтерфейс повинен надавати можливість додавання інформації про водія, додавання маршрутного листа за задану дату у окремому вікні. У головному вікні – вибір водія зі списку та введення періоду розрахунку а також показ плати за перевезення за період.

24.22. Скласти програму з графічним інтерфейсом для обчислення стипендії.

Інформація про студента включає прізвище, ім'я, рік народження, курс та оцінки у останню сесію. Описати клас Student як нащадок класу Person (див. завдання 24.18). Інформація про студентів зберігається у файлі. Надати можливість введення інформації про нового студента у окремому вікні. У головному вікні показувати список студентів та список студентів з нарахованою стипендією.

24.23. Скласти програму з графічним інтерфейсом для розв'язання задачі.

У робочий книзі MS Excel містяться дані про авіарейси. На аркуші «Авіакомпанії» - назва та код авіакомпанії, наприклад

Id	Name
PS	Ukraine International Airlines
AF	Air France

На аркуші «Аеропорти» - код та назва аеропорту, а також назва міста, наприклад:

Id	Airport	City
KBP	Бориспіль	Київ
IEV	Київ	Київ
CDG	Шарль-де-Голь	Париж

На аркуші «Рейси» - інформація про рейси, наприклад:

from_id	to_id	Flight	Days	Depart	Arrive	Class	Cost
KBP	CDG	AF2268	1030060	10:35	13:50	E	8350
IEV	CDG	PS765	0204060	10:35	13:50	E	7500

де Days – номери днів тижня, коли здійснюється рейс (0 означає, що у цей день рейс не здійснюється)

Графічний інтерфейс повинен надавати можливість вибрати перший та другий аеропорт, знайти та показати всі рейси з першого до другого аеропорту на задану дату а також вартість білетів та вибрати рейс.

24.24. В умовах завдання 24.23 показати розклад вильоту рейсів з заданого аеропорту на задану дату.

24.25. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Велика компанія веде список зареєстрованих постачальників для здійснення закупівель. Кожен постачальник має номенклатуру продукції, яку він постачає, та рейтинг у балах на підставі попередніх закупівель.

Дані зберігаються у робочій книзі MS Excel.

На аркуші «Постачальники» зберігається код постачальника, назва, рейтинг, наприклад:

Id	Name	Rating	Adress
----	------	--------	--------

S01	Доміно	4	domino@com.ua
S02	Кондор	6	condor@com.ua

На аркуші «Продукція» зберігається код товару та назва, наприклад:

Id	Name
P01	Олівець
P02	Ручка кулькова

На аркуші «Ціна» зберігається код товару, код постачальника, ціна та термін поставки у днях, наприклад:

S_id	P_id	Price	Term
S01	P01	2,5	5
S02	P01	2,4	6

Нехай, у тендері на закупівлі для відбору найкращого постачальника вибір по кожному товару здійснюється за найбільшим значенням, що отримується з формули:

$$S = a_1 * P / P_{\max} + a_2 * R / R_{\max},$$

де a_1 , a_2 - коефіцієнти, що в сумі дають 1, P – ціна товару у даного постачальника, P_{\max} - максимальна ціна даного товару, R – рейтинг даного постачальника, R_{\max} - максимальний рейтинг усіх постачальників.

Скласти програму, яка для заданого переліку товарів відбирає найкращого постачальника для кожного товару при заданих коефіцієнтах a_1 , a_2 .

Графічний інтерфейс повинен надавати можливість додати товар до переліку товарів у тендері а також вказати кількість одиниць товару. У головному вікні треба показати список товарів з відібраними постачальниками та вартістю кожної позиції.

24.26. Скласти програму з графічним інтерфейсом для розв'язання задачі.

У робочій книзі MS Excel зберігається інформація про замовлення товарів у продавця.

На аркуші «Покупці» міститься інформація про покупців, наприклад:

Id	Name	Address
C01	Доміно	domino@com.ua
C02	Кондор	condor@com.ua

На аркуші «Товари» міститься код та назва товару а також одиниця виміру та ціна, наприклад:

id	Name	Unit	Price
P01	Олівець	шт.	2,5
P02	Ручка кулькова	шт.	2,4

На аркуші «Рахунки» міститься код, номер та дата рахунку а також код покупця, наприклад:

id	No	Date	Client
I01	253	18.07.2016	C01
I02	255	19.07.2016	C02

На аркуші «Пункти» міститься інформація про пункти рахунку. А саме, код рахунку код товару, кількість, наприклад:

I_id	P_id	Quantity
I01	P01	200
I01	P02	150

Скласти програму, яка готує рахунок. Забезпечити графічний інтерфейс додавання рахунку а також додавання або видалення пункту вибраного рахунку.

24.27. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Компанія виконує великий проект, у рамках якого на багатьох майданчиках компанії-замовника паралельно виконуються схожі роботи. Необхідно автоматизувати друк актів виконаних робіт.

Інформація про виконання робіт зберігається у робочій книзі MS Excel.

На аркуші «Майданчики» міститься інформація про код, назву, адресу майданчика та прізвища відповідальних осіб від замовника та виконавця, наприклад:

id	Name	Address	Responsible	Manager
S01	Рубежанська філія ТТС	Рубіжне, вул. Шевченка, 22	Іваненко І.І.	Костенко К.К.
S02	Перечинська філія ТТС	Перечин, вул. Шкільна, 14	Петренко П.П.	Костенко К.К.

На аркуші «Роботи» міститься інформація про код роботи та назву роботи, наприклад:

id	Name
W01	Прокладання кабелю 25 м
W02	Тестування обладнання

На аркуші «Акти» міститься інформація про код акту, номер акту, дату акту, суму за актом, код майданчику, наприклад:

id	No	Date	Sum	S_id
A01	34	18.07.2016	17580	S01
A02	75	19.07.2016	23250	S02

На аркуші «Пункти» міститься інформація про код роботи та код акту, наприклад:

W_id	A_id
W01	A01

W02	A01

Графічний інтерфейс повинен забезпечити можливість формування переліку робіт по вибраному майданчику а також показ вже сформованого переліку.

24.28. Скласти програму з графічним інтерфейсом для проведення тестування. Питання можуть підтримувати вибір одного або декількох варіантів з багатьох альтернатив. Питання та відповіді зберігаються у файлі з позначенням правильного варіанту (ів) відповіді. Програма повинна читати питання, виводити їх на екран, отримувати та зберігати відповіді. Після завершення тестування – аналізувати відповіді та повідомляти кількість правильних відповідей.

24.29. В умовах завдання 24.28 забезпечити контроль часу проходження тесту, вважаючи, що максимальний час також зберігається у файлі.

24.30. В умовах завдання 24.28 забезпечити навчальний режим тестування, надавши можливість після проходження тесту подивитись правильні відповіді на кожне питання а також надані відповіді.

24.31. Скласти програму з графічним інтерфейсом для заповнення простих форм. Форма зберігається у вигляді тексту у текстовому файлі та має поля для заповнення, виділені наступним чином: $\{\$<name> <width>\}$, де name – назва поля, width – розмір поля.

Програма повинна виводити форму у вікно тексту так, щоб замість полів показувати width пропусків. Поля повинні підсвічуватись іншим кольором фону. Окрім цього, програма повинна дозволити введення символів тільки у поля та не більше вказаного розміру поля width. Як результат, програма повинна повернути словник з ключами – назвами полів – та значеннями – введеними рядками.

24.32. Скласти програму з графічним інтерфейсом для перегляду файлів програм у Python. Забезпечити підсвічування різними кольорами ключових слів, рядків, коментарів та іншого тексту.

24.33. Скласти програму з графічним інтерфейсом для перегляду файлів MS Excel. Забезпечити вибір та відкриття файлу, вибір робочого аркуша з наявних робочих аркушів, виведення даних цього робочого аркуша.

24.34. Скласти програму з графічним інтерфейсом для перегляду тексту з файлів MS Word. Забезпечити вибір та відкриття файлу. Також максимально зберегти шрифт, колір шрифту та написання з файлу MS Word.

24.35. Скласти програму з графічним інтерфейсом для порівняння 2 «майже однакових» текстових файлів. Програма повинна виводити вміст 2 файлів, забезпечити їх сумісний перегляд та підсвічувати кольором текст, який є в одному файлі та якого немає у другому файлі, і навпаки.

24.36. Скласти програму з графічним інтерфейсом для реалізації простого калькулятора. Калькулятор повинен мати набір кнопок для введення цифр та арифметичних дій а також вікно для виведення результату як на рисунку нижче.



24.37. Скласти програму з графічним інтерфейсом для розв'язання задачі «Ханойські вежі». Дошка має три стержні. На першому нанизані n дисків спадного вгору діаметра. Потрібно, перекладаючи диски по одному, розташувати їх в колишньому порядку на другому стержні, використовуючи третій стержень для тимчасового зберігання дисків. При цьому більший диск ніколи не повинен розташовуватися над меншим.

Показати анімацію переміщення при заданій кількості дисків.

24.38. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Задані n точок площини. Скласти програму вибору трьох з них таких, щоб периметр трикутника з вершинами в цих точках був найбільшим.

Завдання точок здійснити у графічному режимі на полотні (натисненням у потрібній позиції лівої клавіші миші). Побудований трикутник також показати у графічному режимі.

24.39. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Задані n точок площини. Провести коло мінімального радіусу так, щоб усі точки лежали всередині кола.

Завдання точок здійснити у графічному режимі на полотні (натисненням у потрібній позиції лівої клавіші миші). Побудоване коло також показати у графічному режимі.

24.40. Скласти програму з графічним інтерфейсом для показу аналогового годинника. Показати циферблат, секундну, хвилинну та годинну стрілки. Здійснити анімацію руху стрілок.

24.41. Скласти програму з графічним інтерфейсом для показу піщаного годинника. Показати корпус годинника та область, заповнену піском. Показати анімацію переміщення піску з верхньої до нижньої частини годинника. Дати можливість встановити час вичерпання піску а також перегорнути годинник після вичерпання часу для нового обліку.

24.42. Скласти програму з графічним інтерфейсом для показу коливань фізичного маятника. Дати можливість встановити довжину маятника а також початкове відхилення. Показати анімацію руху маятника.

24.43. Нехай задано шахову позицію. Скласти програму з графічним інтерфейсом для відображення усіх можливих ходів за білих та за чорних.

Підказка: використати клас GridCanvas з матеріалів лекцій. Описати клас BoundText для зображення тексту у клітинках поля. Використати символи Unicode для шахових фігур з таблиці нижче.

Вид	Числовий код	Опис
-----	--------------	------

	#9818;	Чорний король
	#9819;	Чорний ферзь
	#9820;	Чорна тура
	#9821;	Чорний слон
	#9822;	Чорний кінь
	#9823;	Чорний пішак
	#9812;	Білий король
	#9813;	Білий ферзь
	#9814;	Біла тура
	#9815;	Білий слон
	#9816;	Білий кінь
	#9817;	Білий пішак

24.44. Скласти програму з графічним інтерфейсом для відображення шляху шахової фігури «кінь» з поля шахової дошки (x, m) на поле (y, k), де x, y - вертикалі (позначаються літерами від a до h), m, k - горизонталі (позначаються цифрами від 1 до 8). Показувати обчислення шляху та безпосередньо перехід коня по клітинках дошки. Дати можливість вибрати режим: анімація або покроковий.

Підказка: використати клас GridCanvas з матеріалів лекцій. Описати клас BoundText для зображення тексту у клітинках поля. Використати символ Unicode для шахового коня.

24.45. Нехай задано позицію у грі у шашки. Скласти програму з графічним інтерфейсом для відображення усіх можливих ходів за білих та за чорних. Шашки позначати білими та чорними колами з контрастною границею.

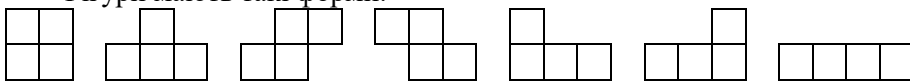
Підказка: використати клас GridCanvas з матеріалів лекцій.

24.46. Скласти програму з графічним інтерфейсом для гри у «Тетрис». Гра полягає у наступному: є «склянка» розміром 20x10, у яку згори з центру падають з постійною швидкістю випадкові фігури різної форми з 4 клітинок.

Гравець керує грою за допомогою клавіш-стрілок. Стрілки вліво та вправо відповідно переміщують фігуру. Стрілка вгору повертає фігуру на 90 градусів. Стрілка вниз «скидає» фігуру до низу.

Якщо деякий ряд поля повністю заповнений клітинками фігур, такий ряд знімається та гравцю нараховуються бали. Гра закінчується, коли чергова фігура відразу після появи не може рухатись до низу. Задача – набрати максимальну кількість балів до закінчення гри.

Фігури мають такі форми:



Підказка: використати клас GridCanvas з матеріалів лекцій. Описати клас BoundRect для зображення прямокутників у клітинках поля.

24.47. Скласти програму з графічним інтерфейсом для реалізації простого графічного редактора. Надати можливість зображувати лінії, прямокутники, еліпси заданого кольору та з заданою товщиною та кольором границі.

Зберігати параметри зображення у файлі та відновлювати з файлу.

24.48. Скласти програму з графічним інтерфейсом для реалізації розставлення меблів у кімнаті. Задавати розміри кімнати. Кожен елемент меблів зображувати прямокутником заданих розмірів та назвою. Меблі можна пересувати та повертати на заданий кут. Меблі не повинні перетинатися під час переміщення. Зберігати параметри зображення кімнати та меблів у файлі та відновлювати з файлу.

24.49. Скласти програму з графічним інтерфейсом для реалізації гри «Посадка на Місяць». Зобразити ламаною лінією рельєф поверхні Місяця. На поверхні повинно бути 2 майданчика для посадки (рівна поверхня шириною не менше ширини корабля). Корабель починає рух до поверхні із заданою вертикальною швидкістю. Швидкість та позиція корабля керуються стрілками вниз (увімкнення двигуна та зменшення швидкості) а також праворуч та ліворуч. Вертикальна швидкість при посадці не повинна перевищувати заданої границі. Зобразити

корабель та полум'я двигуна при увімкненні у вигляді зображень (gif).

24.50. Скласти програму з графічним інтерфейсом для реалізації гри «Зенітка». Зенітка повинна уражати літаки, що рухаються із постійною швидкістю з правого до лівого краю полотна. Час виникнення наступного літака – випадковий. Зенітка може стріляти снарядами з визначеною швидкострільністю та з визначеною швидкістю вильоту снарядів. Постріл здійснюється натисненням клавіші «пропуск». Є також можливість встановити кут нахилу зенітки. Літак уражається, якщо зустрічається зі снарядом. За уражений літак гравцю додаються бали. Завдання – уразити якомога більше літаків.

24.51. Скласти програму з графічним інтерфейсом для реалізації гри «Сапер». Гра полягає у наступному. На полі 9x9 у випадкових клітинках розміщено 10 мін. Натиснення на клітинці лівою клавішею миші «відкриває» клітинку. Якщо у клітинці міна, вона «підривається» та гра закінчується. Якщо ж у клітинці немає міни, то відкриваються усі клітинки до клітинок, що межують з мінами. У таких клітинках показують кількість сусідніх мін по вертикалі горизонталі та діагоналі. Натиснення правої клавіші миші відмічає клітинку як зайняту міною.

Завдання – відкрити всі клітинки, у яких немає мін.

Для мін та прапорців використати зображення (gif).



Підказка: використати клас GridCanvas з матеріалів лекцій. Описати клас BoundText для зображення тексту у клітинках поля та BoundImage для зображення мін та відмічених клітинок.

24.52. Скласти програму з графічним інтерфейсом для реалізації гри у «Морський бій». Гра полягає у наступному: гравець та комп'ютер на полі 10x10 розміщують «кораблі». Кожен корабель – це одна, дві, три або чотири сусідніх клітинки по горизонталі або вертикалі. На полі розміщують 4 кораблі по 1 клітинці, 3 – по 2 клітинки, 2 – по 3 клітинки та 1 – по 4 клітинки. Два кораблі не можуть бути розміщені поруч по горизонталі, вертикалі та діагоналі.

Гравець та комп'ютер по черзі роблять свої ходи. Кожний хід – це вказання клітинки, наприклад A1. Якщо у вказаній клітинці є корабель, то корабель вважається «пораним» (або «вбитим», якщо поранено усі клітинки) та гравець (або комп'ютер) отримує бонусний хід. Виграє той, хто першим уразив усі кораблі опонента.

Графічний інтерфейс повинен показувати ігрове поле гравця та комп'ютера, надавати можливість гравцю розміщувати

кораблі на полі з дотриманням правил а також показувати хід гри (зроблені постріли та підбиті кораблі).



Підказка: використати клас GridCanvas з матеріалів лекцій. Описати клас BoundText для зображення тексту у клітинках поля та BoundRect для зображення прямокутників у клітинках.

24.53. Скласти програму з графічним інтерфейсом для реалізації зображення орієнтованого графу. Вершини зображувати колами, а дуги – лініями зі стрілками. Навантаження вершин показувати біля вершин текстом. Підібрати відображення так, щоб вершини та дуги не зливались. Якщо потрібно, використати лінійки прокрутки на полотні.

24.54. Скласти програму з графічним інтерфейсом для реалізації зображення бінарного дерева. Вершини зображувати колами, а дуги – лініями. Навантаження вершин показувати біля вершин текстом. Підібрати відображення так, щоб вершини та дуги не зливались. Корінь дерева повинен бути згори. Якщо потрібно, використати лінійки прокрутки на полотні.

24.55. Скласти програму з графічним інтерфейсом для реалізації зображення зв'язків між персоналом проектів.

У аркуші робочої книги MS Excel містяться дані про проекти та осіб, задіяних у цих проектах у формі:

Project	Person

--	--

Скласти програму, яка у окремому робочому аркуші буде список зв'язків між особами у формі:

Person1	Person2	Projects	Weight

де Проекти (Projects) – це всі проекти через підкреслення, на яких спільно працювали дві особи, а вага (Weight) зв'язку, - кількість спільних проектів двох осіб.

Показати умовно персонал проектів та існуючі зв'язки, враховуючи їх вагу. Кожну особу показувати колом. Діаметр кола повинен відповідати кількості проектів, у яких особа брала участь: більше проектів – більше діаметр. Зв'язки повинні зображуватись лініями. Товщина ліній відповідає вазі зв'язку: більше вага – більше товщина. Прізвища осіб розміщувати всередині кіл. Підібрати відображення так, щоб кола та лінії не зливались. Якщо потрібно, використати лінійки прокрутки на полотні.

24.56. Скласти програму з графічним інтерфейсом, яка вводить 2 числа у 2 полях введення, що розташовано по вертикалі.

По натисненню кнопки «Порівняти» порівнює 2 введених числа та показує між полями введення відповідне відношення (<, >, =)

24.57. Описати клас TextBolder, який реалізує графічний інтерфейс, що складається з вікна тексту, лінійок прокрутки та кнопки «Bold» нижче вікна тексту. Встановити у вікні тексту шрифт 'Arial' розміром 16.

При натисненні на кнопку Bold, якщо вибраний користувачем текст (якщо є) має зображуватись напівгрубим шрифтом того ж розміру та назви.

Підказка: використати стандартні константи для позначення початку та кінця виділення SEL_FIRST, SEL_LAST

24.58. В умовах попереднього завдання здійснювати зняття напівгрубого шрифту, якщо користувач вибрав (в точності) вже

виділений фрагмент тексту, який зображено напівгрубим шрифтом, та натиснув кнопку Bold.

Підказка: використати методи для вікна тексту txt:

а) txt.tag_ranges(tag_name) – повертає список діапазонів, які помічені тегом tag_name у вигляді [s1, e1, s2, e2, ...], де s1 – початкова позиція, а e1 – кінцева позиція відповідного діапазону

б) txt.compare(pos1, op, pos2) – повертає булівський результат відношення op між позиціями у текстовому вікні pos1 та pos2. Відношення задається рядком '<', '=', тощо.

24.59. У робочому аркуші MS Excel записано відомості про автомобілі у форматі:

марка	модель	VIN	власник	номер держреєстрації

де VIN – унікальний номер кузова автомобіля

Скласти програму з графічним інтерфейсом, яка виконує такі функції:

- додати відомості про автомобіль
- змінити відомості про автомобіль за заданим VIN
- видалити відомості про автомобіль за заданим VIN
- знайти всі автомобілі даної марки
- знайти всі автомобілі даної моделі
- знайти всі автомобілі даного власника
- знайти автомобіль за заданим VIN

Тема 25 Загальна будова глобальних мереж

.....

25.1. Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт читає з текстового файлу та відправляє на сервер рядки, що містять дати у форматі dd.mm.yyyy або yyyy-mm-dd або mm/yyyy/dd. Сервер у кожному рядку приводить дати

до єдиного формату dd.mm.yyyу та відправляє рядок назад клієнту. Клієнт зберігає рядки у новому файлі.

25.2. Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт відправляє на сервер рядки, які містять арифметичні вирази, що складаються з чисел, знаків операцій, дужок. Наприклад, $2 - 57*(33 + 25/4)$. Сервер перевіряє синтаксичну правильність виразу та повертає відповідь клієнту.

25.3. Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт читає з текстового файлу та відправляє на сервер рядки, що містять переписку декількох осіб електронною поштою. Сервер складає список контактів та після завершення передачі повертає цей список клієнту у вигляді рядка, розділеного символами '\n'.

25.4. Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт спочатку відправляє на сервер шаблон регулярного виразу. Потім клієнт читає з текстового файлу та відправляє на сервер рядки. Сервер для кожного рядка перевіряє, чи відповідає рядок заданому шаблону, та надсилає відповідь клієнту.

25.5. Скласти програми сервера та клієнта для розв'язання такої задачі. Установа збирає дані про студентів різних навчальних закладів. Дані надаються клієнтами по мережі по окремих рядках. Для кожного студента вказують прізвище, ім'я, по-батькові, номер студентського квитка, дату народження, телефон. Сервер повинен зберегти всі дані у єдиному текстовому файлі. Також перевірити правильність наданої інформації.

У даних дата може бути у форматі не тільки dd.mm.yyyу, уyyу-mm-dd або mm/yyуу/ dd. Також день та/або місяць може містити одну цифру, а не 2. Також у багатьох записах номер телефону зазначений без вказання + або зовсім без коду країни. Всі телефони необхідно привести до єдиного формату.

Якщо рядок з даними містить правильну інформацію, сервер повертає клієнту рядок '\n'. Якщо формат даних у рядку неправильний, сервер повертає цей рядок клієнту.

25.6. Скласти програми сервера та клієнта з обміну файлами. Обмін ініціює клієнт. Сервер приймає файл та зберігає його у визначеному каталозі.

25.7. В умовах завдання 25.6 зробити так, щоб ініціювати обмін файлами міг як клієнт, так і сервер.

25.8. В умовах завдання 25.6 зробити так, щоб файлами обмінювались два клієнти, а сервер організував та підтримував процес обміну.

25.9. Створити програму, яка здійснює backup файлів заданих каталогів на сервері. Клієнт ініціює backup а сервер зберігає файли із збереженням структури каталогів.

25.10. В умовах завдання T25.9 зробити так, щоб файли для backup надсилав один клієнт, а приймав та зберігав інший. Сервер повинен організувати та підтримувати процес обміну.

25.11. Створити програму (сервер та клієнта), яка здійснює обмін повідомленнями між двома клієнтами у мережі. Можна вважати, що клієнти вводять повідомлення по черзі.

25.12. Створити програму (сервер та клієнта), яка здійснює тестування користувачів у мережі. Усі тестові запитання зберігаються у файлі на сервері. Там же зберігаються правильні відповіді. Кожне питання містить декілька варіантів відповідей, з яких один або декілька можуть бути правильними.

Сервер чекає, поки до нього підключиться певна кількість клієнтів, а потім розпочинає тестування. Кожному клієнту відправляється випадковим чином вибране питання. Клієнт повинен надати відповіді, які він вважає правильними. Сервер аналізує відповіді та у кінці тестування повідомляє клієнтів про кількість набраних балів.

25.13. В умовах завдання 25.12 обмежити час клієнта на надання відповіді.

25.14. В умовах завдання 25.12 передбачити режим «навчання», у якому клієнт може спеціальною командою запитати правильні відповіді на питання.

25.15. В умовах завдання 25.12 сервер повинен записувати результати тестування у журнал. Для цього клієнт повинен надати своє ім'я. Сервер записує у журнал для кожного клієнта дату та час тестування, кількість запитань та кількість набраних балів.

25.16. Скласти програму для гри у «Морський бій» двох гравців у мережі (див. задачу 24.52).

25.17. Скласти програму для гри у шахи двох гравців у мережі (див. задачу 24.43).

25.18. Скласти програму для гри у шашки двох гравців у мережі (див. задачу 24.45).

25.19. Скласти програми сервера та клієнта з синхронізації двох каталогів. Файли з каталогів зберігаються на сервері та клієнті. Після внесення змін склад каталогу клієнта може змінитись (також може бути клієнт на якому застаріла інформація з цього каталогу. Синхронізація повинна усунути розбіжності у двох каталогах. Тобто на сервері та клієнті повинні зберігатись ті ж самі файли з тією ж самою датою/часом модифікації. Синхронізацію ініціює клієнт.

25.20. В умовах завдання 25.19 передбачити 3 режими синхронізації: зі сторони клієнта, зі сторони сервера або двостороння. При синхронізації зі сторони клієнта оновлення відбувається тільки на сервері. Аналогічно працює синхронізація зі сторони сервера: оновлення відбуваються на клієнті.

25.21. Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт відправляє на сервер рядки по 1 рядку. Сервер перевіряє, чи складається рядок з 2 непорожніх симетричних підрядків. Повертає клієнту відповідний рядок відповіді.

Клієнт показує цей рядок відповіді.

Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт відправляє на сервер рядки, які містять присвоєння вигляду:

$$x = e$$

де x – ім'я змінної, e – арифметичний вираз

25.22. Сервер перевіряє синтаксичну правильність виразу, обчислює результат присвоєння та повертає 2 рядки: у першому – оцінка правильності виразу, у другому – значення змінної, якщо вираз правильний. Сервер також має запам'ятовувати обчислені значення змінних та потім використовувати їх у обчисленнях наступних присвоєнь.

Тема 26 Побудова веб-клієнтів

26.1. Скласти програму, яка читає прогноз погоди у заданому місті з сайту sinoptik.ua та зберігає у файлі Excel у окремому рядку поточну дату та прогнози максимальної та мінімальної температури на кожний з наступних 5 днів.

Запит на погоду у заданому місті:

<https://sinoptik.ua/погода-<місто>>

Наприклад,

<https://sinoptik.ua/погода-киев>

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.2. В умовах задачі 26.1, додати інформацію про поточну погоду у файл Excel. Використовуючи актуальні та прогнозні дані, набрати статистику та встановити точність температурного прогнозу на 1, 2, 3, 4, 5 днів.

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.3. Скласти програму, яка читає прогноз погоди у заданому місті з сайту meteoprog.ua та зберігає у файлі Excel у окремому рядку поточну дату та прогнози максимальної та мінімальної температури на кожний з наступних 5 днів.

Запит на погоду у заданому місті:

<http://www.meteoprog.ua/ru/weather/<місто>/>

Наприклад,

<http://www.meteoprog.ua/ru/weather/Kyiv/>

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.4. Скласти програму, яка читає заголовки новин на сайті www.pravda.com.ua за задану дату та виділяє теми або особистостей, які згадуються найчастіше. Для отримання

заголовків новин за дату *dd.mm.yyyy* застосовують запит http://www.pravda.com.ua/news/date_ddmmyyyy/

Вважати, що теми та особистості – це слова, що починаються з великої літери.

Блок заголовку новин у файлі HTML складається з рядків виду:

```
"<div class="article_header">...>заголовок</a>"
```

Потрібний заголовок розташовано замість ‘заголовок’ перед завершальним тегом .

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.5. Скласти програму, яка читає та показує гороскоп на сьогодні за заданим знаком зодіаку з сайту <https://orakul.com/horoscope/astrologic/more/<знак>/today.html>.

Знак – це знак Зодіаку у латинській транскрипції.

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.6. Скласти програму, яка читає з сайту кафедри математичної фізики (<http://matfiz.univ.kiev.ua>) та завантажує у заданий каталог усі приклади програм із заданої теми.

Сторінка з адресами тем - <http://matfiz.univ.kiev.ua/pages/13>

Посилання на сторінку теми на сторінці з адресами має вигляд:

```
<a href=[url-шлях]/pages/YY" >Тема XX
```

де

[url-шлях] – відносний шлях до сторінки (по відношенню до <http://matfiz.univ.kiev.ua>)

YY – номер сторінки

XX – номер теми

Посилання на приклади на сторінці теми має вигляд:

```
a href="/userfiles/files/[файл]"
```

де

[файл] – ім’я файлу прикладу.

Файли прикладів можуть мати розширення .ру або .руw

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.7. Скласти програму, яка буде розклад поїздів з заданого міста України. Okремо виділити розклади прибуття та відправлення поїздів.

Для отримання розкладу використати сайт <https://gd.tickets.ua>. Використати запит:

<https://gd.tickets.ua/uk/railwaytracker/table/~<код станції експрес 3>>

де <код станції експрес 3>можна отримати зі сторінки Вікіпедії за запитом

[https://uk.wikipedia.org/wiki/Список_залізничних_станцій_і_поз'їздів_України_\(<перша_літера_назви_міста>\)](https://uk.wikipedia.org/wiki/Список_залізничних_станцій_і_поз'їздів_України_(<перша_літера_назви_міста>))

Наприклад, для Києва відповідний запит на отримання сторінки з кодами -

[https://uk.wikipedia.org/wiki/Список_залізничних_станцій_і_поз'їздів_України_\(К\)](https://uk.wikipedia.org/wiki/Список_залізничних_станцій_і_поз'їздів_України_(К))

а запит на отримання розкладу, -

<https://gd.tickets.ua/uk/railwaytracker/table/~2200001>

Слід зазначити, що у HTML-сторінці сайту <https://gd.tickets.ua> неправильно вказано кодування. Кодування, яке слід застосовувати, - “windows-1251”.

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.8. Завантажити з сайту мех-мату (<http://mechmat.knu.ua/>) розклад на поточний семестр. Розклад міститься у Excel-файлі.

Треба показати розклад занять групи на заданий день тижня.

Розклад розташовано на сторінці за адресою <http://www.mechmat.knu.ua/golovna/fakul-tet/raspisanie/>

26.9. Скласти програму, яка аналізує статті (колонки) заданого автора за заданий період на сайті <http://www.pravda.com.ua>. Статті доступні за посиланням http://www.pravda.com.ua/columns/page_<xxx>/, де xxx – номер сторінки. Перевірити, чи мають ці статті позитивну чи негативну спрямованість. Про спрямованість статті свідчить використання

слів: при позитивній спрямованості кількість позитивних слів суттєво перевищує кількість негативних. Використати словник, що складається з позитивних та негативних слів.

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

26.10. Скласти програму, яка аналізує новини з сайтів новин за заданий період та обчислює кількість згадувань політиків з заданого списку, впорядковуючи список за незростанням. Для прикладу розглянути сайт <http://www.pravda.com.ua>.

а) Використати регулярні вирази.

б) Використати структурний аналіз HTML та клас HTMLParser.

Тема 27 Побудова веб-серверів

.....

27.1. Скласти програму, яка працює в оточенні веб-сервера, для введення у браузері x , ϵ та обчислення суми усіх доданків, які за абсолютною величиною не менші ϵ :

$$y = \frac{1}{(1+x)^2} = 1 - 2*x + 3*x^2 - \dots, \quad (|x| < 1);$$

Проконтролювати виконання умови $|x| < 1$. Показати результат обчислень на окремій сторінці.

а) використати CGI

б) використати WSGI

27.2. Скласти програму, яка працює в оточенні веб-сервера, для введення рядка та перевірки, чи є цей рядок паліндромом. Показати результат перевірки.

а) використати CGI

б) використати WSGI

27.3. Скласти програму, яка працює в оточенні веб-сервера, для введення рядка та виведення всіх різних слів цього рядка. Слова виводити на окремій сторінці.

- а) використати CGI
- б) використати WSGI

27.4. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Задана непорожня послідовність ненульових цілих чисел, за якою йде 0. Визначити кількість змін знаку в цій послідовності. Наприклад, у послідовності 1, -34, 8, 14, -5, 0 знак змінюється три рази.

Для введення елементів послідовності використати одне поле введення на сторінці. Кнопка «Обробити» повинна ініціювати обробку введеного елементу послідовності та очищення поля введення.

Показати результат обчислень.

- а) використати CGI
- б) використати WSGI

27.5. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Виключити з заданого рядка групи символів, які знаходяться між '(' та ')'. Самі дужки теж мають бути виключені. Вважається, що дужки розставлено правильно (парами) та всередині кожної пари дужок немає інших дужок.

Ввести рядок у браузері та оказати результат обчислень.

- а) використати CGI
- б) використати WSGI

27.6. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Знайти у даному рядку символ та довжину найдовшої послідовності однакових символів, що йдуть підряд.

Ввести рядок у браузері. Показати рядок, виділивши знайдену найдовшу послідовність іншим кольором тексту.

- а) використати CGI
- б) використати WSGI

27.7. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Вивести на друк усі цифри, які входять у заданий рядок, та окремо - решту символів, зберігаючи при цьому взаємне розташування символів у кожній з цих двох груп.

Ввести рядок у браузері та показати 2 рядки-результати.

а) використати CGI

б) використати WSGI

27.8. Скласти програму, яка працює в оточенні веб-сервера, для введення у браузері вектору з n компонент. Перевіряти, чи всі поля заповнено. З використанням цього класу побудувати програму для розв'язання задачі: ввести у браузері вектор з n дійсних компонент та знайти кількість компонент, що належать відрізьку $[a, b]$. Вектор вводити на окремій сторінці. На початковій сторінці ввести n , a , b . Після введення вектору показати компоненти введеного вектору а також результат обчислень.

а) використати CGI

б) використати WSGI

27.9. В умовах завдання 27.8 використати програму для введення у браузері вектору для розв'язання задачі: ввести два вектори з n дійсних компонент та обчислити їх скалярний добуток. Вектори вводити на окремих сторінках. На початковій сторінці ввести n . Показати компоненти введених векторів а також результат обчислень.

а) використати CGI

б) використати WSGI

27.10. Скласти програму, яка працює в оточенні веб-сервера, для введення у браузері матриці $n \times m$. Перевіряти, чи всі поля заповнено. З використанням цієї програми розв'язати задачу: ввести 2 матриці та обчислити їх добуток. Матриці вводити на окремих сторінках. На початковій сторінці ввести розміри матриць. Показати введені матриці та матрицю, що є результатом.

а) використати CGI

б) використати WSGI

27.11. Скласти програму, яка працює в оточенні веб-сервера, для введення у браузері коефіцієнтів поліному степні n . Поля введення для нульових коефіцієнтів можна не заповнювати. З використанням цього класу розв'язати задачу: ввести 2 поліноми

P1, P2 та рядок, який містить вираз, що залежить від 2 поліномів.
Наприклад,

$$P1 + P2 * P1 - P2$$

Обчислити поліном, який буде значенням цього виразу.

Поліноми вводити на окремих сторінках. На початковій сторінці вводити степені поліномів та рядок виразу. Показати введені поліноми та поліном-результат.

- а) використати CGI
- б) використати WSGI

27.12. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Дано файл, який зберігається на сервері та містить відомості про іграшки: вказується назва іграшки (наприклад: м'яч, лялька, конструктор і т.д.), її вартість в гривнях та вікові границі дітей, для яких іграшка призначається (наприклад, для дітей від двох до п'яти років). Підібрати усі іграшки за заданим віком дитини та/або обмеженням вартості.

Вводити відомості про іграшки треба на окремій сторінці. На іншій сторінці вводити обмеження та показувати відібрані іграшки.

- а) використати CGI
- б) використати WSGI

27.13. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Дано файл, який містить відомості про книги. Відомості про кожну книгу - це прізвище автора, назва та рік видання. Підібрати усі книги за заданою назвою та/або автором та/або періодом видання.

Вводити відомості про книги треба на окремій сторінці. На іншій сторінці вводити обмеження та показувати відібрані книги.

- а) використати CGI
- б) використати WSGI

27.14. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

У двох файлах містяться результати футбольного турніру. У першому файлі записано назви команд. У другому файлі – результати матчів у вигляді:

команда1 команда2 m1 m2

де, *команда1, команда2* – номери першої та другої команди;
m1, m2 – кількість м'ячів, забитих відповідно першою та другою командою.

За перемогу нараховується 3 очки, за нічию – 1, за поразку – 0.

З двох команд, які мають однакову кількість очок, першою вважається та, яка

- 1) має кращу різницю забитих та пропущених м'ячів;
- 2) при однаковій різниці має більше забитих м'ячів;
- 3) при всіх однакових попередніх показниках визначається жеребкуванням (використати для жеребкування генератор випадкових чисел).

Показати поточну таблицю турніру у вигляді:

місце, команда, ігор, вигравів, нічиїх, поразок, м'ячів забито, м'ячів пропущено, очок

Програма повинна надавати можливість введення результату матчу на окремій сторінці. На іншій сторінці повинна відображатись поточна таблиця турніру.

- а) використати CGI
- б) використати WSGI

27.15. Скласти програму, яка працює в оточенні веб-сервера, для конвертування валют. Поточні курси валют містяться у файлі Excel на сервері у форматі: <код валюти 1> <код валюти 2> <курс>. Код валюти – це рядок з 3 символів, наприклад UAH, USD, EUR тощо.

Currency1	Currency2	Rate
UAH	USD	24,81
UAH	EUR	27,75
...

Програма повинна забезпечити вибір двох валют зі списків кодів валют на сторінці, введення суми у першій валюті та показ суми у другій валюті.

- а) використати CGI
- б) використати WSGI

27.16. працює в оточенні веб-сервера, для телефонного довідника. Дані про знайомих та їх телефони зберігаються у файлі на сервері. Програма повинна забезпечити додавання інформації про знайомого, редагування інформації про знайомого та пошук телефону за прізвищем.

- а) використати CGI
- б) використати WSGI

27.17. Скласти програму, яка використовує WSGI, для введення таблицю та обчислення заробітної платні співробітників.

Клас Person описано наступним чином:

```
class Person:                                     #Клас Особа
    def __init__(self):
        self.name = None                         #прізвище
        self.byear = None                       #рік народження

    def input(self):                             #ввести особу
        self.name = input('Прізвище: ')
        self.byear = input('Рік народження: ')

    def print(self):                             #вивести особу
        print(self.name, self.byear, end = ' ')
```

Описати клас Співробітник на базі класу Person.

У цьому класі повинно бути як мінімум два додаткових поля: «табельний номер» та «заробітна платня», а також методи введення та виведення інформації про співробітника та розрахунку заробітної платні за кількістю відпрацьованих годин та загальною кількістю годин на місяць.

Використати цей клас для розрахунку заробітної платні за місяць, якщо співробітник працює на умовах погодинної оплати. Для розрахунку вводити відпрацьовані години по днях місяця (табелі). Вважається, що відома кількість годин, яку повинен відпрацювати співробітник кожного дня, для отримання 100% платні.

Передбачити дії: введення таблицю для співробітника, введення таблиць всіх співробітників, розрахунку заробітної платні всіх співробітників.

Табелі зберігати у файлі на сервері та читати з файлу.

Вводити таблицю для співробітника на окремій сторінці. На іншій сторінці показувати список співробітників та список співробітників з їх заробітною платнею.

27.18. Скласти програму, яка використовує WSGI, для розрахунку плати за проживання.

Описати клас Гість на базі класу Person (див. завдання 27.17) та клас ТипНомера з полями «назва типу», «плата за добу».

У класі Гість повинно бути як мінімум два додаткових поля: «номер» та «кількість днів проживання», а також методи введення та виведення інформації про гостя та розрахунку плати за проживання.

Використати цей клас для реєстрації гостя у номері та розрахунку плати за проживання усіх гостей.

Номери та інформацію про гостей зберігати у файлах на сервері та читати з файлів.

Програма повинна передбачати реєстрацію гостя на окремій сторінці. При реєстрації вибрати номер зі списку номерів та вказати кількість днів проживання. На іншій сторінці показувати список гостей та плату за проживання.

27.19. Скласти програму, яка використовує WSGI, для розрахунку плати за проїзд.

Описати клас Пасажир на базі класу Person (див. завдання 27.17).

У класі Пасажир повинно бути як мінімум два додаткових поля: «місто відправлення» та «місто прибуття», а також методи введення та виведення інформації про пасажирів та розрахунку плати за білет.

Використати цей клас для розрахунку плати за білети усіх пасажирів. Вважати що маршрути зберігаються у списку кортежів (місто1, місто2, відстань), а також те, що плата за білет пропорційна відстані та відома плата за 1 км відстані. Маршрути зберігаються у файлі.

Надати можливість для введення нового маршруту на окремій сторінці а також для додавання пасажирів на окремій сторінці. На іншій сторінці вивести список маршрутів, дати можливість вибрати маршрут, пасажирів та обчислити та показати вартість проїзду пасажирів. Зберегти дані пасажирів у файлі на сервері.

27.20. Скласти програму, яка використовує WSGI, для розрахунку плати за перевезення.

Описати клас Водій на базі класу Person (див. завдання 27.17).

У класі Водій повинно бути як мінімум два додаткових поля: «плата за тоно-кілометр» та «вантажопідйомність автомобіля», а також методи введення та виведення інформації про водія та розрахунку плати за перевезення вантажу на задану відстань.

Маршрутний лист водія за певну дату містить кількість кілометрів перевезень вантажу.

Використати цей клас для розрахунку плати всіх водіїв за період. Вважати що маршрутні листи водіїв зберігаються у списку кортежів (водій, дата, відстань), а також те, що плата за перевезення пропорційна добутку відстані на вагу вантажу. Дані водіїв зберігаються у файлі на сервері.

Програма повинна надавати можливість додавання інформації про водія, додавання маршрутного листа за задану дату на окремій сторінці. На іншій сторінці – вибір водія зі списку та введення періоду розрахунку а також показ плати за перевезення за період.

27.21. Скласти програму, яка використовує WSGI, для обчислення стипендії.

Інформація про студента включає прізвище, ім'я, рік народження, курс та оцінки у останню сесію. Описати клас Student як нащадок класу Person (див. завдання 27.17). Інформація про студентів зберігається у файлі. Надати можливість введення інформації про нового студента на окремій сторінці. На іншій сторінці показувати список студентів та список студентів з нарахованою стипендією.

27.22. Скласти програму, яка використовує WSGI, для розв'язання задачі.

У робочий книзі MS Excel на сервері містяться дані про авіарейси. На аркуші «Авіакомпанії» - назва та код авіакомпанії, наприклад

Id	Name
PS	Ukraine International Airlines
AF	Air France

--	--

На аркуші «Аеропорти» - код та назва аеропорту, а також назва міста, наприклад:

Id	Airport	City
KBP	Бориспіль	Київ
IEV	Київ	Київ
CDG	Шарль-де-Голль	Париж

На аркуші «Рейси» - інформація про рейси, наприклад:

from_Id	to_Id	Flight	Days	Depart	Arrive	Class	Cost
KBP	CDG	AF2268	1030060	10:35	13:50	E	8350
IEV	CDG	PS765	0204060	10:35	13:50	E	7500

де Days – номери днів тижня, коли здійснюється рейс (0 означає, що у цей день рейс не здійснюється)

Програма повинна надавати можливість вибрати перший та другий аеропорт, знайти та показати всі рейси з першого до другого аеропорту на задану дату а також вартість білетів та вибрати рейс.

27.23. В умовах завдання 27.22 показати розклад вильоту рейсів з заданого аеропорту на задану дату.

27.24. Скласти програму, яка використовує WSGI, для розв’язання задачі.

Велика компанія веде список зареєстрованих постачальників для здійснення закупівель. Кожен постачальник має номенклатуру продукції, яку він постачає, та рейтинг у балах на підставі попередніх закупівель.

Дані зберігаються у робочій книзі MS Excel на сервері.

На аркуші «Постачальники» зберігається код постачальника, назва, рейтинг, наприклад:

Id	Name	Rating	Adress
S01	Доміно	4	domino@com.ua
S02	Кондор	6	condor@com.ua

На аркуші «Продукція» зберігається код товару та назва, наприклад:

Id	Name
P01	Олівець
P02	Ручка кулькова

На аркуші «Ціна» зберігається код товару, код постачальника, ціна та термін поставки у днях, наприклад:

S_id	P_id	Price	Term
S01	P01	2,5	5
S02	P01	2,4	6

Нехай, у тендері на закупівлі для відбору найкращого постачальника вибір по кожному товару здійснюється за найбільшим значенням, що отримується з формули:

$$S = a_1 * P / P_{\max} + a_2 * R / R_{\max},$$

де a_1 , a_2 - коефіцієнти, що в сумі дають 1, P – ціна товару у даного постачальника, P_{\max} - максимальна ціна даного товару, R – рейтинг даного постачальника, R_{\max} - максимальний рейтинг усіх постачальників.

Скласти програму, яка для заданого переліку товарів відбирає найкращого постачальника для кожного товару при заданих коефіцієнтах a_1 , a_2 .

Програма повинна надавати можливість додати товар до переліку товарів у тендері а також вказати кількість одиниць товару. На іншій сторінці треба показати список товарів з відібраними постачальниками та вартістю кожної позиції.

27.25. Скласти програму, яка використовує WSGI, для розв'язання задачі.

У робочій книзі MS Excel на сервері зберігається інформація про замовлення товарів у продавця.

На аркуші «Покупці» міститься інформація про покупців, наприклад:

Id	Name	Address
C01	Доміно	domino@com.ua
C02	Кондор	condor@com.ua

На аркуші «Товари» міститься код та назва товару а також одиниця виміру та ціна, наприклад:

id	Name	Unit	Price
P01	Олівець	шт.	2,5
P02	Ручка кулькова	шт.	2,4

На аркуші «Рахунки» міститься код, номер та дата рахунку а також код покупця, наприклад:

id	No	Date	Client
I01	253	18.07.2016	C01
I02	255	19.07.2016	C02

На аркуші «Пункти» міститься інформація про пункти рахунку. А саме, код рахунку код товару, кількість, наприклад:

I_id	P_id	Quantity
I01	P01	200
I01	P02	150

Скласти програму, яка готує рахунок. Забезпечити додавання рахунку а також додавання або видалення пункту вибраного рахунку у браузері.

27.26. Скласти програму, яка використовує WSGI, для розв'язання задачі.

Компанія виконує великий проект, у рамках якого на багатьох майданчиках компанії-замовника паралельно виконуються схожі роботи. Необхідно автоматизувати друк актів виконаних робіт.

Інформація про виконання робіт зберігається у робочій книзі MS Excel на сервері.

На аркуші «Майданчики» міститься інформація про код, назву, адресу майданчика та прізвища відповідальних осіб від замовника та виконавця, наприклад:

Id	Name	Address	Responsible	Manager
S01	Рубежанська філія ТТС	Рубіжне, вул. Шевченка, 22	Іваненко І.І.	Костенко К.К.
S02	Перечинська філія ТТС	Перечин, вул. Шкільна, 14	Петренко П.П.	Костенко К.К.

На аркуші «Роботи» міститься інформація про код роботи та назву роботи, наприклад:

Id	Name
W01	Прокладання кабелю 25 м
W02	Тестування обладнання

На аркуші «Акти» міститься інформація про код акту, номер акту, дату акту, суму за актом, код майданчику, наприклад:

id	No	Date	Sum	S_id
A01	34	18.07.2016	17580	S01
A02	75	19.07.2016	23250	S02

На аркуші «Пункти» міститься інформація про код роботи та код акту, наприклад:

W_id	A_id
W01	A01
W02	A01

Програма повинна забезпечити можливість формування переліку робіт по вибраному майданчику а також показ вже сформованого переліку у браузері.

27.27. Скласти програму, яка працює в оточенні веб-сервера, виводить форму для введення цілого числа, та перевіряє, чи є це число факторіалом деякого натурального числа.

Тема 28 XML та JSON

.....

28.1. Скласти програму, яка працює в оточенні веб-сервера, для введення рядка та повернення всіх різних слів цього рядка. Слова повертати у форматі

- а) JSON
- б) XML

Структуру даних JSON та XML визначити самостійно

28.2. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Знайти у даному рядку символ та довжину найдовшої послідовності однакових символів, що йдуть підряд.

Ввести рядок у браузері. Повернути символ та довжину найдовшої послідовності символів у форматі:

- а) JSON
- б) XML

Структуру даних JSON та XML визначити самостійно

28.3. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Повернути усі цифри, які входять в заданий рядок, та окремо - решту символів, зберігаючи при цьому взаємне розташування символів у кожній з цих двох груп.

Ввести рядок у браузері та повернути 2 рядки-результати у форматі

- а) JSON
- б) XML

Структуру даних JSON та XML визначити самостійно

28.4. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Дано файл у форматі JSON, який зберігається на сервері та містить відомості про іграшки: вказується назва іграшки (наприклад: м'яч, лялька, конструктор і т.д.), її вартість в гривнях та вікові границі дітей, для яких іграшка призначається (наприклад, для дітей від двох до п'яти років). Підібрати усі іграшки за заданим віком дитини та/або обмеженням вартості.

Вводити відомості про іграшки треба на окремій сторінці. На іншій сторінці вводити обмеження. Програма повинна повертати відібрані іграшки у форматі JSON.

Структуру даних JSON визначити самостійно

28.5. В умовах завдання 28.4 замінити формат JSON на формат XML.

28.6. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Дано файл у форматі

а) JSON,

б) XML,

який містить відомості про книги. Відомості про кожену книгу - це прізвище автора, назва та рік видання. Підібрати усі книги за заданою назвою та/або автором та/або періодом видання.

Вводити відомості про книги треба на окремій сторінці. На іншій сторінці вводити обмеження та показувати відібрані книги.

28.7. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

У двох файлах у форматі

а) JSON

б) XML

містяться результати футбольного турніру. У першому файлі записано назви команд. У другому файлі – результати матчів у вигляді:

команда1 команда2 m1 m2

де, *команда1, команда2* – номери першої та другої команди;

m1, m2 – кількість м'ячів, забитих відповідно першою та другою командою.

За перемогу нараховується 3 очки, за нічию – 1, за поразку – 0.

З двох команд, які мають однакову кількість очок, першою вважається та, яка

- 1) має кращу різницю забитих та пропущених м'ячів;
- 2) при однаковій різниці має більше забитих м'ячів;
- 3) при всіх однакових попередніх показниках визначається жеребкуванням (використати для жеребкування генератор випадкових чисел).

Показати поточну таблицю турніру у вигляді:

місце, команда, ігор, виграшів, нічиїх, поразок, м'ячів забито, м'ячів пропущено, очок

Програма повинна надавати можливість введення результату матчу на окремій сторінці. На іншій сторінці повинна відображатись поточна таблиця турніру.

28.8. Скласти програму, яка працює в оточенні веб-сервера, для конвертування валют. Поточні курси валют містяться у файлі

а) JSON

б) XML

на сервері у форматі: <код валюти 1> <код валюти 2> <курс>.

Код валюти – це рядок з 3 символів, наприклад UAH, USD< EUR тощо.

Currency1	Currency2	Rate
UAH	USD	24,81
UAH	EUR	27,75
...

Програма повинна забезпечити вибір двох валют зі списків кодів валют на сторінці, введення суми у першій валюті та показ суми у другій валюті.

28.9. Скласти програму, яка використовує WSGI, для введення таблицю та обчислення заробітної платні співробітників.

Клас Person описано наступним чином:

```
class Person:                                     #Клас Особа
    def __init__(self):
        self.name = None                          #прізвище
```



```

self.byear = None           #рік народження

def input(self):           #ввести особу
    self.name = input('Прізвище: ')
    self.byear = input('Рік народження: ')

def print(self):          #вивести особу
    print(self.name, self.byear, end = ' ')

```

Описати клас Співробітник на базі класу Person.

У цьому класі повинно бути як мінімум два додаткових поля: «табельний номер» та «заробітна платня», а також методи введення та виведення інформації про співробітника та розрахунку заробітної платні за кількістю відпрацьованих годин та загальною кількістю годин на місяць.

Використати цей клас для розрахунку заробітної платні за місяць, якщо співробітник працює на умовах погодинної оплати. Для розрахунку вводити відпрацьовані години по днях місяця (табель). Вважається, що відома кількість годин, яку повинен відпрацювати співробітник кожного дня, для отримання 100% платні.

Передбачити дії: введення таблицю для співробітника, введення таблиць всіх співробітників, розрахунку заробітної платні всіх співробітників.

Табель зберігати у файлі

- а) JSON
- б) XML

на сервері та читати з файлу.

Вводити таблиць для співробітника на окремій сторінці. На іншій сторінці показувати список співробітників та список співробітників з їх заробітною платнею.

28.10. Скласти програму, яка використовує WSGI, для розрахунку плати за проживання.

Описати клас Гість на базі класу Person (див. завдання 28.9) та клас ТипНомера з полями «назва типу», «плата за добу».

У класі Гість повинно бути як мінімум два додаткових поля: «номер» та «кількість днів проживання», а також методи введення та виведення інформації про гостя та розрахунку плати за проживання.

Використати цей клас для реєстрації гостя у номері та розрахунку плати за проживання усіх гостей.

Номери та інформацію про гостей зберігати у файлах

а) JSON

б) XML

на сервері та читати з файлів.

Програма повинна передбачати реєстрацію гостя на окремій сторінці. При реєстрації вибрати номер зі списку номерів та вказати кількість днів проживання. На іншій сторінці показувати список гостей та плату за проживання.

28.11. Скласти програму, яка використовує WSGI, для розрахунку плати за проїзд.

Описати клас Пасажир на базі класу Person (див завдання 28.9).

У класі Пасажир повинно бути як мінімум два додаткових поля: «місто відправлення» та «місто прибуття», а також методи введення та виведення інформації про пасажера та розрахунку плати за білет.

Використати цей клас для розрахунку плати за білети усіх пасажирів. Вважати що маршрути зберігаються у списку кортежів (місто1, місто2, відстань), а також те, що плата за білет пропорційна відстані та відома плата за 1 км відстані. Маршрути зберігаються у файлі у форматі JSON.

Надати можливість для введення нового маршруту на окремій сторінці а також для додавання пасажера на окремій сторінці. На іншій сторінці вивести список маршрутів, дати можливість вибрати маршрут, пасажера та обчислити та показати вартість проїзду пасажера. Зберегти дані пасажирів у файлі у форматі JSON на сервері.

28.12. В умовах завдання 28.11 замінити формат JSON на формат XML.

28.13. Скласти програму, яка використовує WSGI, для розрахунку плати за перевезення.

Описати клас Водій на базі класу Person (див. завдання 28.9).

У класі Водій повинно бути як мінімум два додаткових поля: «плата за тоно-кілометр» та «вантажопідйомність автомобіля», а

також методи введення та виведення інформації про водія та розрахунку плати за перевезення вантажу на задану відстань.

Маршрутний лист водія за певну дату містить кількість кілометрів перевезень вантажу.

Використати цей клас для розрахунку плати всіх водіїв за період. Вважати що маршрутні листи водіїв зберігаються у списку кортежів (водій, дата, відстань), а також те, що плата за перевезення пропорційна добутку відстані на вагу вантажу. Дані водіїв зберігаються у файлах

а) JSON

б) XML

на сервері.

Програма повинна надавати можливість додавання інформації про водія, додавання маршрутного листа за задану дату на окремій сторінці. На іншій сторінці – вибір водія зі списку та введення періоду розрахунку а також показ плати за перевезення за період.

28.14. Скласти програму, яка використовує WSGI, для обчислення стипендії.

Інформація про студента включає прізвище, ім'я, рік народження, курс та оцінки у останню сесію. Описати клас Student як нащадок класу Person (див. завдання 28.9). Інформація про студентів зберігається у файлі

а) JSON

б) XML

Надати можливість введення інформації про нового студента на окремій сторінці. На іншій сторінці показувати список студентів та список студентів з нарахованою стипендією.

28.15. Скласти програму, яка використовує WSGI, для розв'язання задачі.

У файлах JSON на сервері містяться дані про авіарейси. У файлі з відомостями про авіакомпанії - назва та код авіакомпанії, наприклад

Id	Name
PS	Ukraine International Airlines
AF	Air France

--	--

У файлі з відомостями про аеропорти - код та назва аеропорту, а також назва міста, наприклад:

Id	Airport	City
KBP	Бориспіль	Київ
IEV	Київ	Київ
CDG	Шарль-де-Голль	Париж

У файлі з відомостями про рейси - інформація про рейси, наприклад:

from_Id	to_Id	Flight	Days	Depart	Arrive	Class	Cost
KBP	CDG	AF2268	1030060	10:35	13:50	E	8350
IEV	CDG	PS765	0204060	10:35	13:50	E	7500

де Days – номери днів тижня, коли здійснюється рейс (0 означає, що у цей день рейс не здійснюється)

Програма повинна надавати можливість вибрати перший та другий аеропорт, знайти та показати всі рейси з першого до другого аеропорту на задану дату а також вартість білетів та вибрати рейс.

28.16. В умовах завдання 28.15 замінити формат JSON на формат XML.

28.17. В умовах завдання 28.15 показати розклад вильоту рейсів з заданого аеропорту на задану дату. Розглянути файли у форматі

- a) JSON
- б) XML

28.18. Скласти програму, яка використовує WSGI, для розв'язання задачі.

Велика компанія веде список зареєстрованих постачальників для здійснення закупівель. Кожен постачальник має номенклатуру продукції, яку він постачає, та рейтинг у балах на підставі попередніх закупівель.

Дані зберігаються у файлах JSON на сервері.

У файлі з відомостями про постачальників зберігається код постачальника, назва, рейтинг, наприклад:

Id	Name	Rating	Adress
S01	Доміно	4	domino@com.ua
S02	Кондор	6	condor@com.ua

У файлі з відомостями про продукцію зберігається код товару та назва, наприклад:

Id	Name
P01	Олівець
P02	Ручка кулькова

У файлі з відомостями про ціну зберігається код товару, код постачальника, ціна та термін поставки у днях, наприклад:

S_id	P_id	Price	Term
S01	P01	2,5	5
S02	P01	2,4	6

Нехай, у тендері на закупівлі для відбору найкращого постачальника вибір по кожному товару здійснюється за найбільшим значенням, що отримується з формули:

$$S = a_1 * P / P_{\max} + a_2 * R / R_{\max},$$

де a_1 , a_2 - коефіцієнти, що в сумі дають 1, P – ціна товару у даного постачальника, P_{\max} - максимальна ціна даного товару, R – рейтинг даного постачальника, R_{\max} - максимальний рейтинг усіх постачальників.

Скласти програму, яка для заданого переліку товарів відбирає найкращого постачальника для кожного товару при заданих коефіцієнтах a_1 , a_2 .

Програма повинна надавати можливість додати товар до переліку товарів у тендері а також вказати кількість одиниць

товару. На іншій сторінці треба показати список товарів з відібраними постачальниками та вартістю кожної позиції.

28.19. В умовах завдання 28.18 замінити формат JSON на формат XML.

28.20. Скласти програму, яка використовує WSGI, для розв'язання задачі.

У файлах JSON на сервері зберігається інформація про замовлення товарів у продавця.

У файлі з відомостями про покупців міститься інформація про покупців, наприклад:

Id	Name	Address
C01	Доміно	domino@com.ua
C02	Кондор	condor@com.ua

У файлі з відомостями про товари міститься код та назва товару а також одиниця виміру та ціна, наприклад:

Id	Name	Unit	Price
P01	Олівець	шт.	2,5
P02	Ручка кулькова	шт.	2,4

У файлі з відомостями про рахунки міститься код, номер та дата рахунку а також код покупця, наприклад:

Id	No	Date	Client
I01	253	18.07.2016	C01
I02	255	19.07.2016	C02

У файлі з відомостями про пункти міститься інформація про пункти рахунку. А саме, код рахунку код товару, кількість, наприклад:

I_id	P_id	Quantity
I01	P01	200

I01	P02	150

Скласти програму, яка готує рахунок. Забезпечити додавання рахунку а також додавання або видалення пункту вибраного рахунку у браузері.

28.21. В умовах завдання 28.20 замінити формат JSON на формат XML.

28.22. Скласти програму, яка використовує WSGI, для розв'язання задачі.

Компанія виконує великий проект, у рамках якого на багатьох майданчиках компанії-замовника паралельно виконуються схожі роботи. Необхідно автоматизувати друк актів виконаних робіт.

Інформація про виконання робіт зберігається у файлах JSON на сервері.

У файлі з відомостями про майданчики міститься інформація про код, назву, адресу майданчика та прізвища відповідальних осіб від замовника та виконавця, наприклад:

Id	Name	Address	Responsible	Manager
S01	Рубежанська філія ТТС	Рубіжне, вул. Шевченка, 22	Іваненко І.І.	Костенко К.К.
S02	Перечинська філія ТТС	Перечин, вул. Шкільна, 14	Петренко П.П.	Костенко К.К.

У файлі з відомостями про роботи міститься інформація про код роботи та назву роботи, наприклад:

Id	Name
W01	Прокладання кабелю 25 м
W02	Тестування обладнання

У файлі з відомостями про акти міститься інформація про код акту, номер акту, дату акту, суму за актом, код майданчику, наприклад:

Id	No	Date	Sum	S_id
A01	34	18.07.2016	17580	S01
A02	75	19.07.2016	23250	S02

У файлі з відомостями про пункти міститься інформація про код роботи та код акту, наприклад:

W_id	A_id
W01	A01
W02	A01

Програма повинна забезпечити можливість формування переліку робіт по вибраному майданчику а також показ вже сформованого переліку у браузері.

28.23. В умовах завдання 28.22 замінити формат JSON на формат XML.

28.24. Скласти програму, яка використовує API з сервера <http://openweathermap.org/api> для визначення поточних погодних умов у заданому місті або за географічними координатами.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі:

- a) JSON
- б) XML

28.25. Скласти програму, яка використовує API з сервера <http://openweathermap.org/api> для визначення прогнозу погоди на 5 днів у заданому місті або за географічними координатами.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі:

- a) JSON
- б) XML

28.26. Скласти програму, яка використовує API з сервера <https://www.wunderground.com/weather/api> для визначення

поточних погодних умов у заданому місті або за географічними координатами.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі:

- а) JSON
- б) XML

28.27. Скласти програму, яка використовує API з сервера <https://www.wunderground.com/weather/api> для визначення прогнозу погоди на 10 днів у заданому місті або за географічними координатами.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі:

- а) JSON
- б) XML

28.28. Скласти програму, яка використовує API з сервера <https://www.apixu.com/api.aspx> для визначення поточних погодних умов у заданому місті або за географічними координатами або за IP-адресою.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі:

- а) JSON
- б) XML

28.29. Скласти програму, яка використовує API з сервера <https://www.apixu.com/api.aspx> для визначення прогнозу погоди на 10 днів у заданому місті або за географічними координатами або за IP-адресою.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі:

- а) JSON
- б) XML

28.30. Скласти програму, яка використовує API з сервера <https://www.apixu.com/api.aspx> для побудови графіку максимальної та мінімальної температури за визначений місяць

визначеного року (2015 або 2016) у заданому місті або за географічними координатами або за IP-адресою.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі:

а) JSON

б) XML

28.31. Скласти програму, яка використовує API з сервера <https://sandbox.amadeus.com/api-catalog> для пошуку країн у які можна вилетіти з заданого аеропорту у заданий діапазон дат за задану максимальну ціну.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі JSON.

28.32. Скласти програму, яка використовує API з сервера <https://sandbox.amadeus.com/api-catalog> для пошуку найкращого з точки зору ціни польоту дня вильоту з заданого аеропорту у заданий аеропорт у заданий діапазон дат та враховуючи задану тривалість подорожі.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі JSON.

28.33. Скласти програму, яка використовує API з сервера <https://sandbox.amadeus.com/api-catalog> для пошуку найкращого з точки зору ціни польоту рейсу з заданого аеропорту у заданий аеропорт у задану дату та враховуючи задану тривалість подорожі.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі JSON.

28.34. Скласти програму, яка використовує API з сервера електронних декларацій <https://declarations.com.ua/api/> для пошуку усіх декларантів, які заробили за рік суму, більшу заданої. Опис документа JSON, який повертає сервер, розташований за адресою https://github.com/dchaplinsky/declarations.com.ua/blob/master/declarations_site/catalog/data/mapping_defs.json

28.35. Скласти програму, яка використовує API з сервера даних Світового Банку <https://datahelpdesk.worldbank.org/knowledgebase/topics/125589-developer-information> для побудови графіку обсягу валового внутрішнього продукту по роках за заданий період для заданих країн. Дані отримувати у форматі

- а) JSON
- б) XML

28.36. Скласти програму, яка використовує API з сервера даних Світового Банку <https://datahelpdesk.worldbank.org/knowledgebase/topics/125589-developer-information> для знаходження 10 країн з найбільшою варіацією валового внутрішнього продукту на душу населення по роках за заданий період. Варіацію обчислювати як суму модулів різниць показника за сусідні роки. Дані отримувати у форматі

- а) JSON
- б) XML

28.37. Скласти програму, яка використовує API з сервера відкритих даних Верховної Ради України <http://opendata.rada.gov.ua/?q=api> для знаходження 10 народних депутатів, які задекларували за один рік найбільшу кількість транспортних засобів. Опис відповідного набору даних розміщено за адресою <http://opendata.rada.gov.ua/?q=node/840/api>. Дані отримувати у форматі

- а) JSON
- б) XML

28.38. Скласти програму, яка використовує API з сервера <http://products.wolframalpha.com/api/> для знаходження інтегралу стандартної функції на заданому відрізьку. Використати Wolfram|Alpha Full Results API.

Для використання API потрібно зареєструватись на сервері та отримати API key.

Отримати та проаналізувати дані у форматі

- а) JSON
- б) XML

Підказка: пошуковий запит може бути у довільній формі але починати слід зі слова “integrate”, за яким вказати функцію та відрізок.

28.39. Скласти програму, що читає слова з заданого документу у мережі (url) та зберігає їх разом з кількістю входжень кожного слова у файлі у форматі XML. Слова розділяються пропусками та розділовими знаками.

Структура XML

```
<words>
```

```
<word count="кількість" > слово </word>
```

```
...
```

```
</words>
```

28.40. Скласти програму, що працює в оточенні веб-сервера, отримує від клієнта url, виконує завдання з попередньої задачі та повертає клієнту XML.

28.41. Скласти програму, що читає слова з заданого текстового файлу та зберігає їх разом з кількістю входжень кожного слова у файлі у форматі JSON. Слова розділяються пропусками та розділовими знаками.

Структура JSON:

```
[{"word": <слово>, "count": <кількість>}, ...]
```

Тема 29 Використання баз даних

.....

29.1. Скласти програму для роботи з базою даних, що містить дати народження знайомих. Програма повинна реалізовувати функції додавання знайомого, показу дати народження за прізвищем. При запуску програма повинна показувати дати народження знайомих, для яких до дня народження залишилось не більше 7 днів.

29.2. Скласти програму для роботи з базою даних, що містить логіни та паролі входу до систем. Для кожної системи у БД зберігається неформальна назва, адреса (якщо є), логін та пароль.

Реалізувати функції додавання системи та повернення адреси, логіну та пароллю за назвою системи.

29.3. Скласти програму для роботи з базою даних, що містить означення понять та їх опис. Реалізувати функції додавання поняття та повернення опису за введеним поняттям.

29.4. Скласти програму для роботи з базою даних, що містить комунальні платежі. У БД зберігаються назви компаній, неформальні назви платежів а також рахунки від компаній разом з відомостями про оплату. Для кожного рахунку вказують: компанію, неформальну назву платежу, рік та місяць оплати, дату рахунку, суму рахунку, дату оплати, суму оплати. Реалізувати функції додавання компанії, додавання неформальної назви платежу, додавання рахунку, сплати рахунку, показу усіх рахунків за заданий місяць та рік а також загальної суми рахунків та сплаченої суми.

29.5. Скласти програму для роботи з базою даних, що містить інформацію про постачальників товару. Для кожного постачальника вказано його назву та контактні дані. У окремих таблицях БД зберігаються дані про товари а також дані про постачальників товарів. Реалізувати функції додавання постачальника, додавання товару, фіксації факту, що постачальник постачає певний товар а також пошуку за назвою товару усіх постачальників, що постачають товар та пошуку за назвою постачальника усіх товарів, що постачає постачальник.

29.6. Скласти програму для роботи з базою даних, що містить інформацію про власний сад. У БД зберігається інформація про рід дерев (яблуна, груша тощо), сорт дерев, рік посадки, неформальне місце посадки. Для кожного дерева зберігають також врожай по роках. Реалізувати функції додавання роду дерев, додавання сорту, додавання дерева, додавання врожаю за заданий рік, повернення інформації про всі дерева даного роду та про врожай заданого дерева за заданий період років.

29.7. Скласти програму для роботи з базою даних, що містить рецепти страв. У окремих таблицях БД зберігати дані про інгредієнти а також про назви страв. Рецепт складається з переліку інгредієнтів разом з їх масою (об'ємом) а також з опису рецепту. Для однієї страви може бути декілька рецептів.

Інгредієнти для кожного рецепту та їх масу (об'єм) зберігати у окремій таблиці. Реалізувати функції додавання інгредієнту, додавання страви, додавання рецепту а також пошуку рецептів за назвою страви.

29.8. Скласти програму для роботи з базою даних, що містить інформацію про об'єкти нерухомості. Для кожного об'єкту вказують його вид (будинок, квартира тощо), адресу, загальну площу, кількість кімнат. Окремо зберігається інформація про кожен кімнату: призначення кімнати, площа кімнати. Реалізувати функції додавання об'єкту, додавання кімнати, відбір та показ усієї інформації про об'єкти за заданим видом та площею.

29.9. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Дано базу даних, яка зберігається на сервері та містить відомості про іграшки: вказується назва іграшки (наприклад: м'яч, лялька, конструктор і т.д.), її вартість в гривнях та вікові границі дітей, для яких іграшка призначається (наприклад, для дітей від двох до п'яти років). Підібрати усі іграшки за заданим віком дитини та/або обмеженням вартості.

Вводити відомості про іграшки треба на окремій сторінці. На іншій сторінці вводити обмеження. Програма повинна повертати відібрані іграшки на окремій сторінці.

Структуру бази даних визначити самостійно.

29.10. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Дано базу даних, яка містить відомості про іграшки: вказується назва іграшки (наприклад: м'яч, лялька, конструктор і т.д.), її вартість в гривнях та вікові границі дітей, для яких іграшка призначається (наприклад, для дітей від двох до п'яти років). Підібрати усі іграшки за заданим віком дитини та/або обмеженням вартості.

Вводити відомості про іграшки треба у окремому вікні. У іншому вікні вводити обмеження. Програма повинна повертати відібрані іграшки у окремому вікні.

Структуру бази даних визначити самостійно.

29.11. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Дано базу даних, яка містить відомості про книги. Відомості про кожну книгу - це прізвище автора, назва та рік видання. Підібрати усі книги за заданою назвою та/або автором та/або періодом видання.

Вводити відомості про книги треба на окремій сторінці. На іншій сторінці вводити обмеження та показувати відібрані книги.

29.12. Скласти програму з графічним інтерфейсом для розв'язання задачі.

Дано базу даних, яка містить відомості про книги. Відомості про кожну книгу - це прізвище автора, назва та рік видання. Підібрати усі книги за заданою назвою та/або автором та/або періодом видання.

Вводити відомості про книги треба у окремому вікні. У іншому вікні вводити обмеження та показувати відібрані книги.

Структуру бази даних визначити самостійно.

29.13. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

У базі даних містяться результати футбольного турніру. У одній таблиці записано назви команд. У іншій таблиці – результати матчів у вигляді:

команда1 команда2 m1 m2

де, *команда1, команда2* – номери першої та другої команди; *m1, m2* – кількість м'ячів, забитих відповідно першою та другою командою.

За перемогу нараховується 3 очки, за нічию – 1, за поразку – 0.

З двох команд, які мають однакову кількість очок, першою вважається та, яка

- 1) має кращу різницю забитих та пропущених м'ячів;
- 2) при однаковій різниці має більше забитих м'ячів;
- 3) при всіх однакових попередніх показниках визначається жеребкуванням (використати для жеребкування генератор випадкових чисел).

Показати поточну таблицю турніру у вигляді:

місце, команда, ігор, вигравів, нічиїх, поразок, м'ячів забито, м'ячів пропущено, очок

Програма повинна надавати можливість введення результату матчу на окремій сторінці. На іншій сторінці повинна відображатись поточна таблиця турніру.

Структуру бази даних визначити самостійно.

29.14. Скласти програму, яка працює в оточенні веб-сервера, для конвертування валют. Поточні курси валют містяться у базі даних на сервері. Таблиця курсів валют має вигляд: <код валюти 1> <код валюти 2> <курс>. Код валюти – це рядок з 3 символів, наприклад UAH, USD, EUR тощо.

Currency1	Currency2	Rate
UAH	USD	24,81
UAH	EUR	27,75
...

Програма повинна забезпечити вибір двох валют зі списків кодів валют на сторінці, введення суми у першій валюті та показ суми у другій валюті.

Структуру бази даних визначити самостійно.

29.15. Скласти програму, яка працює в оточенні веб-сервера, для введення таблицю та обчислення заробітної платні співробітників.

Клас Person описано наступним чином:

```
class Person:                                     #Клас Особа
    def __init__(self):
        self.name = None                          #прізвище
        self.byear = None                         #рік народження

    def input(self):                               #ввести особу
        self.name = input('Прізвище: ')
        self.byear = input('Рік народження: ')

    def print(self):                               #вивести особу
        print(self.name, self.byear, end = ' ')
```

Описати клас Співробітник на базі класу Person.

У цьому класі повинно бути як мінімум два додаткових поля: «табельний номер» та «заробітна платня», а також методи

введення та виведення інформації про співробітника та розрахунку заробітної платні за кількістю відпрацьованих годин та загальною кількістю годин на місяць.

Використати цей клас для розрахунку заробітної платні за місяць, якщо співробітник працює на умовах погодинної оплати. Для розрахунку вводити відпрацьовані години по днях місяця (табель). Вважається, що відома кількість годин, яку повинен відпрацювати співробітник кожного дня, для отримання 100% платні.

Передбачити дії: введення таблицю для співробітника, введення таблиців всіх співробітників, розрахунку заробітної платні всіх співробітників.

Табель та дані співробітників зберігати у базі даних на сервері та читати з бази даних.

Вводити таблиць для співробітника на окремій сторінці. На іншій сторінці показувати список співробітників та список співробітників з їх заробітною платнею.

Структуру бази даних визначити самостійно.

29.16. Скласти програму, яка працює в оточенні веб-сервера, для розрахунку плати за проживання.

Описати клас Гість на базі класу Person (див. завдання 29.15) та клас ТипНомера з полями «назва типу», «плата за добу».

У класі Гість повинно бути як мінімум два додаткових поля: «номер» та «кількість діб проживання», а також методи введення та виведення інформації про гостя та розрахунку плати за проживання.

Використати цей клас для реєстрації гостя у номері та розрахунку плати за проживання усіх гостей.

Номери та інформацію про гостей зберігати у базі даних на сервері та читати з бази даних.

Програма повинна передбачати реєстрацію гостя на окремій сторінці. При реєстрації вибрати номер зі списку номерів та вказати кількість діб проживання. На іншій сторінці показувати список гостей та плату за проживання.

Структуру бази даних визначити самостійно.

29.17. Скласти програму, яка працює в оточенні веб-сервера, для розрахунку плати за проїзд.

Описати клас Пасажир на базі класу Person (див завдання 29.15).

У класі Пасажир повинно бути як мінімум два додаткових поля: «місто відправлення» та «місто прибуття», а також методи введення та виведення інформації про пасажирів та розрахунку плати за білет.

Використати цей клас для розрахунку плати за білети усіх пасажирів. Вважати що маршрути зберігаються у списку кортежів (місто1, місто2, відстань), а також те, що плата за білет пропорційна відстані та відома плата за 1 км відстані. Маршрути зберігаються у базі даних на сервері.

Надати можливість для введення нового маршруту на окремій сторінці а також для додавання пасажирів на окремій сторінці. На іншій сторінці вивести список маршрутів, дати можливість вибрати маршрут, пасажирів та обчислити та показати вартість проїзду пасажирів. Зберегти дані пасажирів у базі даних на сервері.

Структуру бази даних визначити самостійно.

29.18. Скласти програму, яка працює в оточенні веб-сервера, для розрахунку плати за перевезення.

Описати клас Водій на базі класу Person (див. завдання 29.15).

У класі Водій повинно бути як мінімум два додаткових поля: «плата за тоно-кілометр» та «вантажопідйомність автомобіля», а також методи введення та виведення інформації про водія та розрахунку плати за перевезення вантажу на задану відстань.

Маршрутний лист водія за певну дату містить кількість кілометрів перевезень вантажу.

Використати цей клас для розрахунку плати всіх водіїв за період. Вважати що маршрутні листи водіїв зберігаються у списку кортежів (водій, дата, відстань), а також те, що плата за перевезення пропорційна добутку відстані на вагу вантажу. Дані водіїв зберігаються у базі даних на сервері.

Програма повинна надавати можливість додавання інформації про водія, додавання маршрутного листа за задану дату на окремій сторінці. На іншій сторінці – вибір водія зі списку та введення періоду розрахунку а також показ плати за перевезення за період.

Структуру бази даних визначити самостійно.

29.19.Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

У бази даних на сервері містяться дані про авіарейси. У таблиці з відомостями про авіакомпанії - назва та код авіакомпанії, наприклад

Id	Name
PS	Ukraine International Airlines
AF	Air France

У таблиці з відомостями про аеропорти - код та назва аеропорту, а також назва міста, наприклад:

Id	Airport	City
KBP	Бориспіль	Київ
IEV	Київ	Київ
CDG	Шарль-де-Голь	Париж

У таблиці з відомостями про рейси - інформація про рейси, наприклад:

from_Id	to_Id	Flight	Days	Depart	Arrive	Class	Cost
KBP	CDG	AF2268	1030060	10:35	13:50	E	8350
IEV	CDG	PS765	0204060	10:35	13:50	E	7500

де Days – номери днів тижня, коли здійснюється рейс (0 означає, що у цей день рейс не здійснюється)

Програма повинна надавати можливість вибрати перший та другий аеропорт, знайти та показати всі рейси з першого до другого аеропорту на задану дату а також вартість білетів та вибрати рейс.

Структуру бази даних визначити самостійно.

29.20. В умовах завдання 29.19 показати розклад вильоту рейсів з заданого аеропорту на задану дату.

29.21. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Велика компанія веде список зареєстрованих постачальників для здійснення закупівель. Кожен постачальник має номенклатуру продукції, яку він постачає, та рейтинг у балах на підставі попередніх закупівель.

Дані зберігаються у базі даних на сервері.

У таблиці з відомостями про постачальників зберігається код постачальника, назва, рейтинг, наприклад:

Id	Name	Rating	Adress
S01	Доміно	4	domino@com.ua
S02	Кондор	6	condor@com.ua

У таблиці з відомостями про продукцію зберігається код товару та назва, наприклад:

Id	Name
P01	Олівець
P02	Ручка кулькова

У таблиці з відомостями про ціну зберігається код товару, код постачальника, ціна та термін поставки у днях, наприклад:

S_id	P_id	Price	Term
S01	P01	2,5	5
S02	P01	2,4	6

Нехай, у тендері на закупівлі для відбору найкращого постачальника вибір по кожному товару здійснюється за найбільшим значенням, що отримується з формули:

$$S = a_1 * P / P_{\max} + a_2 * R / R_{\max},$$

де a_1 , a_2 - коефіцієнти, що в сумі дають 1, P – ціна товару у даного постачальника, P_{\max} - максимальна ціна даного товару, R – рейтинг даного постачальника, R_{\max} - максимальний рейтинг усіх постачальників.

Скласти програму, яка для заданого переліку товарів відбирає найкращого постачальника для кожного товару при заданих коефіцієнтах a_1, a_2 .

Програма повинна надавати можливість додати товар до переліку товарів у тендері а також вказати кількість одиниць товару. На іншій сторінці треба показати список товарів з відібраними постачальниками та вартістю кожної позиції.

Структуру бази даних визначити самостійно.

29.22. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

У базі даних на сервері зберігається інформація про замовлення товарів у продавця.

У таблиці з відомостями про покупців міститься інформація про покупців, наприклад:

Id	Name	Address
C01	Доміно	domino@com.ua
C02	Кондор	condor@com.ua

У таблиці з відомостями про товари міститься код та назва товару а також одиниця виміру та ціна, наприклад:

Id	Name	Unit	Price
P01	Олівець	шт.	2,5
P02	Ручка кулькова	шт.	2,4

У таблиці з відомостями про рахунки міститься код, номер та дата рахунку а також код покупця, наприклад:

Id	No	Date	Client
I01	253	18.07.2016	C01
I02	255	19.07.2016	C02

У таблиці з відомостями про пункти міститься інформація про пункти рахунку. А саме, код рахунку код товару, кількість, наприклад:

I_id	P_id	Quantity
I01	P01	200
I01	P02	150

Скласти програму, яка готує рахунок. Забезпечити додавання рахунку а також додавання або видалення пункту вибраного рахунку у браузері.

Структуру бази даних визначити самостійно.

29.23. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Компанія виконує великий проект, у рамках якого на багатьох майданчиках компанії-замовника паралельно виконуються схожі роботи. Необхідно автоматизувати друк актів виконаних робіт.

Інформація про виконання робіт зберігається у базі даних на сервері.

У таблиці з відомостями про майданчики міститься інформація про код, назву, адресу майданчика та прізвища відповідальних осіб від замовника та виконавця, наприклад:

Id	Name	Address	Responsible	Manager
S01	Рубежанська філія ТТС	Рубіжне, вул. Шевченка, 22	Іваненко І.І.	Костенко К.К.
S02	Перечинська філія ТТС	Перечин, вул. Шкільна, 14	Петренко П.П.	Костенко К.К.

У таблиці з відомостями про роботи міститься інформація про код роботи та назву роботи, наприклад:

Id	Name
W01	Прокладання кабелю 25 м
W02	Тестування обладнання

У таблиці з відомостями про акти міститься інформація про код акту, номер акту, дату акту, суму за актом, код майданчику, наприклад:

Id	No	Date	Sum	S_id
A01	34	18.07.2016	17580	S01
A02	75	19.07.2016	23250	S02

У таблиці з відомостями про пункти міститься інформація про код роботи та код акту, наприклад:

W_id	A_id
W01	A01
W02	A01

Програма повинна забезпечити можливість формування переліку робіт по вибраному майданчику а також показ вже сформованого переліку у браузері.

Структуру бази даних визначити самостійно.

29.24. У файлі MS Excel на 2 робочих аркушах записано дані про робітників підприємства. На першому аркуші – дані про робітників:

- Табельний номер
- Прізвище
- Ім'я
- По батькові
- Паспорт
- Заробітна платня
- Код підрозділу

На другому аркуші – дані про підрозділи:

- Код підрозділу
- Назва підрозділу

Описати програму яка створює базу даних з 2 таблиць для робітників та підрозділів та додає усі дані з MS Excel до цієї бази.

Тема 30 Тестування. Розповсюдження власних застосувань

30.1. Описати функцію, що повертає суму всіх доданків при заданому значенні x , що за абсолютною величиною не перевищують заданого $\varepsilon > 0$.

Скласти програму для тестування цієї функції при декількох значеннях x та ε :

$$\text{а) } y = \ln(1+x) = x - \frac{x^2}{2!} + \frac{x^3}{3!} - \dots, \quad (|x| < 1);$$

$$\text{б) } y = \frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots, \quad (|x| < 1);$$

$$\text{в) } y = \ln \frac{1+x}{1-x} = 2 * \left[\frac{x}{1} + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right], \quad (|x| < 1);$$

$$\text{г) } y = \frac{1}{(1+x)^2} = 1 - 2 * x + 3 * x^2 - \dots, \quad (|x| < 1);$$

$$\text{д) } y = \frac{1}{(1+x)^3} = 1 - \frac{2 * 3}{2} x + \frac{3 * 4}{2} x^2 - \frac{4 * 5}{2} x^3 + \dots, \quad (|x| < 1);$$

$$\text{е) } y = \frac{1}{1+x^2} = 1 - x^2 + x^4 - x^6 + \dots, \quad (|x| < 1);$$

$$\text{є) } y = \sqrt{1+x} = 1 + \frac{1}{2} x - \frac{1}{2 * 4} x^2 + \frac{1 * 3}{2 * 4 * 6} x^3 - \dots, \quad (|x| < 1);$$

$$\text{ж) } y = \frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2} x + \frac{1 * 3}{2 * 4} x^2 - \frac{1 * 3 * 5}{2 * 4 * 6} x^3 - \dots, \quad (|x| < 1);$$

$$\text{з) } y = \arcsin x = x + \frac{1}{2} \frac{x^3}{3!} + \frac{1 * 3}{2 * 4} \frac{x^5}{5!} + \dots, \quad (|x| < 1).$$

30.2. Описати функцію, що повертає

а) суму елементів матриці

- б) мінімальний елемент матриці
- в) максимальний елемент матриці

Скласти програму для тестування цієї функції.

30.3. Дано два рядки однакової довжини: s_1 та s_2 . Відстанню Хемінга між s_1 та s_2 називають кількість позицій, у яких $s_1[i] \neq s_2[i]$. Описати функцію для обчислення відстані Хемінга двох рядків.

Скласти програму для тестування цієї функції.

30.4. Описати функцію для обчислення значення натурального числа за заданим рядком символів, який є записом цього числа у системі числення за основою b ($2 \leq b \leq 16$). Використати функцію, яка за заданим символом повертає відповідну цифру у системі числення за основою b . Використати у цій функції твердження про стан програми `assert` для перевірки того, що відповідний символ є цифрою у системі числення за основою b . Обробити у підпрограмі помилку неправильного символу рядка та показати змістовне повідомлення про помилку.

Скласти програму для тестування цих функцій з урахуванням перевірки обробки виключення при неправильному символі рядка.

30.5. У текстовому файлі записана непорожня послідовність дійсних чисел, які розділяються пропусками в одному рядку та можуть бути розташовані у різних рядках. Визначити функцію обчислення найбільшого з цих чисел.

Забезпечити обробку помилок, якщо у файлі зустрічаються не дійсні числа.

Скласти програму для тестування цієї функції з урахуванням перевірки обробки виключення, якщо зустрівлось не дійсне число.

а) Додатково підготувати тестове оточення (файл)

б) Додатково використати удавані об'єкти та функцію `patch` для заміни дій над файлом.

30.6. Клас `Stack` описано наступним чином:

```
class Stack:
    '''Реалізує стек на базі списку.
    ...
    def __init__(self):
        '''Створити порожній стек.
```

```

'''
self._lst = []           #список елементів стеку

def isempty(self):
    '''Чи порожній стек?.'''
    '''
    return len(self._lst) == 0

def push(self, data):
    '''Вштовхнути елемент у стек.'''
    '''
    self._lst.append(data)

def pop(self):
    '''Взяти елемент зі стеку.'''
    '''
    if self.isempty():
        print('Pop: Стек порожній')
        exit(1)
    data = self._lst.pop()
    return data

```

Описати власний клас на базі Stack, у якому перевизначити метод pop так, щоб він ініціював виключення при спробі взяти елемент порожнього стеку.

Скласти програму для тестування цього класу (усіх його методів) з урахуванням перевірки обробки виключення.

30.7. Клас Черга описано наступним чином:

```

class Queue:
    '''Реалізує чергу на базі списку.'''
    '''
    def __init__(self):
        '''Створити порожню чергу.'''
        '''
        self._lst = []           #список елементів черги

    def isempty(self):
        '''Чи порожня черга?.'''
        '''
        return len(self._lst) == 0

    def add(self, data):
        '''Додати елемент в кінець черги.'''

```

```

    """
    self._lst.append(data)

def take(self):
    """Взяти елемент з початку черги.
    """
    if self.isempty():
        print('Take: Черга порожня')
        exit(1)
    data = self._lst.pop(0)
#перший елемент черги - це нульовий елемент списку
    return data

def __del__(self):
    """Закінчити роботу з чергою.
    """
    print('Deleting queue')
    del self._lst

```

Описати власний клас на базі Queue, у якому перевизначити метод take так, щоб він ініціював виключення при спробі взяти елемент порожньої черги.

Скласти програму для тестування цього класу (усіх його методів) з урахуванням перевірки обробки виключення.

30.8. Клас Кільцевий список описано наступним чином:

```

class Rlist:
    """Реалізує кільцевий список на базі списку.
    """
    def __init__(self):
        """Створити порожній список.
        """
        self._lst = [] #список елементів
        self._cur = None #індекс поточного елемента

    def len(self):
        """Довжина списку.
        """
        return len(self._lst)

    def next(self):
        """Перейти до наступного елемента.
        """
        l = self.len()

```

```

if l != 0:
    if self._cur == l-1:
        #для (l-1) елемента наступним буде нульовий
        self._cur = 0
    else:
        self._cur += 1

def getcurrent(self):
    '''Повернути поточний елемент.
    '''
    if self.len() == 0:
        print('getcurrent: список порожній')
        exit(1)
    data = self._lst[self._cur]
    return data

def update(self, data):
    '''Оновити поточний елемент.
    '''
    if self.len() == 0:
        print('update: список порожній')
        exit(1)
    self._lst[self._cur] = data

def insert(self, data):
    '''Вставити елемент перед поточним.
    '''
    if self.len() == 0:
        #якщо список порожній
        self._lst.append(data)
        #додаємо елемент, він стає поточним
        self._cur = 0
    else:
        self._lst.insert(self._cur,data)
        #інакше вставляємо елемент перед поточним
        self._cur += 1 #щоб поточний елемент
        # не змінився, треба індекс збільшити на 1

def delete(self):
    '''Видалити поточний елемент.
    '''
    if self.len() == 0:
        print('delete: список порожній')
        exit(1)
    l = self.len()
    del self._lst[self._cur]

```

```

if l == 1:
    #якщо список після видалення елемента спорожніє
    self._cur = None
elif self._cur == l-1:
    #якщо поточним був останній елемент списку
    self._cur = 0
    #поточним стане елемент з індексом 0
#else: pass якщо поточним був не останній
# елемент, нічого не робити

def __del__(self):
    '''Закінчити роботу зі списком.
    '''
    del self._lst

```

Описати власний клас на базі Rlist, у якому перевизначити методи `getcurrent`, `update`, `delete` так, щоб вони ініціювали виключення у випадку порожнього списку.

Скласти програму для тестування цього класу (усіх його методів) з урахуванням перевірки обробки виключення.

30.9. Описати клас для роботи з відрізками на числовій осі. Для відрізка передбачити поля:

(a, b, empty)

де a, b - границі відрізка, empty - ознака того, що відрізок порожній.

Реалізувати методи:

- 1) зробити відрізок t порожнім;
- 2) чи порожній відрізок t ;
- 3) покласти відрізок t рівним a, b ;
- 4) покласти відрізок t рівним перетину відрізків $t1, t2$.

Скласти програму для тестування цього класу (усіх його методів).

30.10. Описати клас для реалізації мультимножини цілих чисел на базі словника. Мультимножина - це множина в якій для кожного елемента запам'ятовується не лише його входження, але й кількість входжень.

Кількість входжень елемента k ($0 \leq k \leq n$) у мультимножину - це значення елемента словника з ключем k .

Реалізувати методи:

- 1) зробити мультимножину порожньою;

- 2) чи є мультимножина порожньою;
- 3) додати елемент до мультимножини;
- 4) забрати елемент з мультимножини (кількість входжень елемента зменшується на 1, якщо елемент не входить - відмова);
- 5) кількість входжень елемента у мультимножину;
- 6) об'єднання двох мультимножин (в результаті об'єднання кількість входжень елемента визначається як максимальна з двох мультимножин);
- 7) перетин двох мультимножин (в результаті кількість входжень елемента визначається як мінімальна з двох мультимножин);

Скласти програму для тестування цього класу (усіх його методів).

30.11. У текстовому файлі є дати, задані у форматі dd.mm.yyyу або у форматі уyyу/mm/dd. Також день та/або місяць може містити одну цифру, а не 2. Привести всі дати до єдиного формату dd.mm.yyyу.

Вказівка: використати регулярні вирази та функцію (метод) sub.

Скласти програму для тестування цієї програми.

- а) Додатково підготувати тестове оточення (файл)
- б) Додатково використати удавані об'єкти та функцію patch для заміни дій над файлом.
- в) Додатково використати удавані об'єкти та функцію patch для заміни дій над регулярними виразами.

30.12. За допомогою регулярних виразів розбити текст у текстовому файлі на речення.

Скласти програму для тестування цієї програми.

- а) Додатково підготувати тестове оточення (файл)
- б) Додатково використати удавані об'єкти та функцію patch для заміни дій над файлом.
- в) Додатково використати удавані об'єкти та функцію patch для заміни дій над регулярними виразами.

30.13. У текстовому файлі містяться дати у форматі dd.mm.yyyу або підкреслення для запису дат вручну __.__._____

Знайти всі дати у тексті. Замість підкреслень вставити поточну дату. Зберегти оновлений текст.

Скласти програму для тестування цієї програми.

- а) Додатково підготувати тестове оточення (файл)
- б) Додатково використати удавані об'єкти та функцію patch для заміни дій над файлом.
- в) Додатково використати удавані об'єкти та функцію patch для заміни дій над регулярними виразами.

30.14. Скласти програму порівняння двох каталогів. Програма повинна показувати різницю у каталогах, тобто файли, що присутні в одному каталозі та відсутні у іншому. Якщо файл присутній у одному та іншому каталозі, він не повинен бути показаний. Результат роботи програми слід спрямувати у текстовий файл.

Скласти програму для тестування цієї програми.

- а) Додатково підготувати тестове оточення (файл)
- б) Додатково використати удавані об'єкти та функцію patch для заміни дій над файлом.
- в) Додатково використати удавані об'єкти та функцію patch для заміни дій над каталогами (модуль os).

30.15. Скласти програму порівняння файлів з однаковим ім'ям у двох каталогах. Для таких пар файлів треба показати різницю у часі створення файлу, тобто, який з двох файлів цієї пари був створений пізніше. Якщо файли були створені одночасно, то нічого показувати не потрібно. Результат роботи програми слід спрямувати у текстовий файл.

Скласти програму для тестування цієї програми.

- а) Додатково підготувати тестове оточення (файл)
- б) Додатково використати удавані об'єкти та функцію patch для заміни дій над файлом.
- в) Додатково використати удавані об'єкти та функцію patch для заміни дій над каталогами (модуль os).

30.16. Деяка існуюча програма записує у системний журнал. Системні журнали містяться у заданому каталозі та створюються по декілька файлів на кожну дату. Відповідно, дата та час створення є частиною імені файлу.

Треба скласти програму, яка архівує системні журнали у заданому каталозі за попередні дати у форматі tar.gz.

Скласти програму для тестування цієї програми.

а) Додатково підготувати тестове оточення (файл)

б) Додатково використати удавані об'єкти та функцію patch для заміни дій над файлом.

в) Додатково використати удавані об'єкти та функцію patch для заміни дій над каталогами (модуль os).

30.17. Скласти програму, яка читає прогноз погоди у заданому місті з сайту sinoptik.ua та зберігає у файлі Excel у окремому рядку поточну дату та прогнози максимальної та мінімальної температури на кожний з наступних 5 днів.

Запит на погоду у заданому місті:

<https://sinoptik.ua/погода-<місто>>

Наприклад,

<https://sinoptik.ua/погода-киев>

Скласти програму для тестування цієї програми.

а) Додатково використати удавані об'єкти та функцію patch для заміни дій з читання даних з мережі.

б) Додатково використати удавані об'єкти та функцію patch для заміни дій над файлами MS Excel.

30.18. Скласти програму, яка працює в оточенні веб-сервера, для введення рядка та повернення всіх різних слів цього рядка. Слова повертати у форматі JSON. Структуру даних JSON визначити самостійно.

Скласти програму для тестування цієї програми. Додатково використати удавані об'єкти та функцію patch для заміни дій з читання даних з мережі та повернення відповіді клієнту.

30.19. Скласти програму, яка працює в оточенні веб-сервера, для розв'язання задачі.

Знайти у даному рядку символ та довжину найдовшої послідовності однакових символів, що йдуть підряд.

Повернути символ та довжину найдовшої послідовності символів у форматі XML. Структуру даних XML визначити самостійно.

Скласти програму для тестування цієї програми. Додатково використати удавані об'єкти та функцію patch для заміни дій з читання даних з мережі та повернення відповіді клієнту.

30.20. Скласти програму для роботи з базою даних, що містить логіни та паролі входу до систем. Для кожної системи у БД зберігається неформальна назва, адреса (якщо є), логін та пароль. Реалізувати функції додавання системи та повернення адреси, логіну та паролю за назвою системи.

Скласти програму для тестування цієї програми.

а) Додатково підготувати тестове оточення (базу даних)

б) Додатково використати удавані об'єкти та функцію patch для заміни дій над базою даних.

30.21. Скласти програму для роботи з базою даних, що містить означення понять та їх опис. Реалізувати функції додавання поняття та повернення опису за введеним поняттям.

Скласти програму для тестування цієї програми.

а) Додатково підготувати тестове оточення (базу даних)

б) Додатково використати удавані об'єкти та функцію patch для заміни дій над базою даних.

30.22. Створити структуру каталогів та описати необхідні файли для розповсюдження вибраного власноруч написаного модуля. Підготувати заархівований файл та встановити його у Python з диску.

Тема 31 Паралельні обчислення

.....

31.1. Описати клас-нащадок Thread, який забезпечує передачу необроблених виключень, що виникли у потоці, потоку, який запустив даний. Для цього передбачити окремий метод. Перевірити виконання для потоку обчислення факторіалу, коли аргумент не є цілим числом.

31.2. Описати програму з графічним інтерфейсом, у якому користувач вводить номер n числа Фібоначчі, а програма

викликає рекурсивну функцію обчислення n -го числа Фібоначчі. Функція має виконуватись у окремому потоці. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного числа Фібоначчі і в цьому вікні показувати результат після завершення обчислень.

31.3. Один потік через випадкові інтервали від 1 до t_1 надає повідомлення. Інший потік має обробити ці повідомлення. Обробка займає від 1 до t_2 . Скласти програму, яка генерує та обробляє повідомлення. Повідомлення мають оброблятися у порядку їх надходження. Обробка повідомлення – це просто показ його на екрані. Використати чергу.

31.4. В умовах попередньої задачі вважати, що є пріоритетні повідомлення, які мають бути оброблені раніше, ніж звичайні. Забезпечити таку обробку. Описати власний клас на базі стандартного класу `deque`, який забезпечує розміщення та подальшу обробку повідомлень.

31.5. В умовах задачі 31.3 для обробки повідомлень описати власний клас, у якому використати 2 черги (об'єкти стандартного класу `Queue`).

31.6. Описати програму з графічним інтерфейсом, у якому користувач вводить номер ціле число, а програма викликає функцію що перевіряє, чи є це число простим. Функція має виконуватись у окремому потоці. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного простого числа і в цьому вікні показувати результат після завершення обчислень.

31.7. У супермаркеті є N кас. У деякий момент часу працюють m касирів. У черзі до касирів стоять k покупців. Новий покупець додається до черги до деякої каси через випадковий інтервал від 1 до t_1 . Касир обслуговує покупця за випадковий інтервал від 1 до t_2 . Через випадковий інтервал від 1 до t_3 один з касирів йде на перерву на час від 1 до t_4 . Якщо черга зростає, адміністрація додає резервного касира, до якого переходять покупці з інших черг. Зафіксувати параметри $N, m, k, t_1, t_2, t_3, t_4$. Для цих

параметрів шляхом моделювання розрахувати приблизно потрібну кількість резервних касирів та довжину черги, при якій слід додавати резервного касира, щоб максимальна довжина черги до 1 касира не перевищила l_{\max} .

Вважати, що загальна кількість кас N є достатньою, щоб задовольнити умову. Також вважати, що покупці самостійно рівномірно розподіляються між чергами.

Вказівка: одна черга до каси – це 1 потік.

31.8. Є готель з N номерами. Через випадковий час від 1 до t_1 приходить новий клієнт та заселяється у один з номерів (якщо є вільний), у якому живе час від 1 до t_2 . Якщо вільних номерів немає, то клієнт очікує на звільнення будь-якого номера. При заданих параметрах N , t_1 , t_2 розрахувати середню довжину черги клієнтів та середній час очікування.

Вказівка: один клієнт – це 1 потік.

31.9. Є стадіон на N глядачів з m турнікетами. Турнікети відчиняють за $-t$ до початку матчу. Через 1 турнікет глядач проходить за випадковий час від 1 до t_1 . Усі глядачі приходять до стадіону до початку матчу у випадкові моменти часу але з рівномірним розподілом. Змоделювати проходження глядачів через турнікети та розрахувати, за який мінімальний час до початку матчу треба прийти, щоб з імовірністю 0.9 пройти на стадіон до початку матчу при заданих значеннях параметрів.

Вказівка: один турнікет – це 1 потік.

31.10. Скласти програму, яка має 2 потоки. Основний потік у процесі роботи має створювати у деякому каталозі тимчасові файли з розширенням `.tmp`. Другий потік-демон через певний проміжок часу t_1 перевіряє цей каталог та видаляє файли з розширенням `.tmp`, які були змінені не пізніше ніж за час t_2 до моменту перевірки.

31.11. Залізниця має одноколіїну ділянку. Цю ділянку поїзди проходять за деякий випадковий час у діапазоні від t_1 до t_2 . Якщо у момент підходу зустрічного потяга одноколіїна ділянка зайнята, то зустрічний потяг чекає її звільнення. Промоделювати рух потягів, вважаючи що наступний потяг з кожного напрямку підходить до ділянки через випадковий інтервал від t_3 до t_4 . Кожний потяг – це 1 потік.

Використати черги.

31.12. В умовах задачі 31.11 використати окремі потоки та пул потоків.

31.13. В умовах задачі 31.11 використати блокування.

31.14. Скласти програму, у якій є основний процес та підпроцес. Підпроцес генерує текстові файли великого розміру та додає їх до деякого каталогу Каталог1. У основному процесі головний потік аналізує цей каталог і якщо з'явився новий файл або декілька нових файлів, то ініціює перевірку, чи містить цей файл рядок, що відповідає деякому шаблону. Перевірку здійснюють додаткові потоки. Додатковий потік після перевірки переміщує файли, що містять вказаний шаблон, у Каталог2, а ті, що не містять, - у Каталог3. Використати пул з n додаткових потоків.

31.15. В умовах задачі 31.14 змінити процес роботи наступним чином. Допоміжні потоки працюють постійно. Основний потік, коли з'явився новий файл, додає завдання до черги, яку розбирають додаткові потоки. Використати окремі потоки та чергу.

31.16. Один процес через випадкові інтервали від l до t_1 надає повідомлення. Інший процес має обробити ці повідомлення. Обробка займає від l до t_2 . Скласти програму, яка генерує та обробляє повідомлення. Повідомлення мають оброблятися у порядку їх надходження. Обробка повідомлення – це просто показ його на екрані. Використати чергу.

31.17. Описати програму з графічним інтерфейсом, у якому користувач вводить номер n числа Фібоначчі, а програма викликає рекурсивну функцію обчислення n -го числа Фібоначчі. Функція має виконуватись у окремому процесі. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного числа Фібоначчі і в цьому вікні показувати результат після завершення обчислень.

31.18. Описати програму з графічним інтерфейсом, у якому користувач вводить номер ціле число, а програма викликає функцію що перевіряє, чи є це число простим. Функція має

виконуватись у окремому процесі. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного простого числа і в цьому вікні показувати результат після завершення обчислень.

31.19. Скласти програму, яка має 2 процеси. Основний процес у своїй роботі має створювати у деякому каталозі тимчасові файли з розширенням .tmp. Другий процес через певний проміжок часу t1 перевіряє цей каталог та видаляє файли з розширенням .tmp, які були змінені не пізніше ніж за час t2 до моменту перевірки.

31.20. Скласти програму, яка перевіряє правильність навчальних програм студентів. Кожна студентська програма написана у Python, має певну кількість вхідних даних та певний 1 результат. Вхідні дані вводяться з клавіатури. Результат – це останній рядок, що виводиться на екран. Результатами можуть бути цілі, дійсні числа або рядки (звичайно, що на екран усе виводиться у вигляді рядка).

Створювана програма має запустити студентську програму у окремому процесі, передати їй вхідні дані, почекати визначений максимальний час завершення, проаналізувати отриманий результат. Має бути проаналізовано: чи завершено програму у визначений максимальний час, чи були необроблені виключення під час виконання програми, чи правильно отримано результат (для результатів дійсного типу проаналізувати близькість до очікуваного значення).

Параметри для перевірки містяться у окремому текстовому файлі – у кожному рядку – по 1 завданню. Параметри мають вигляд:

<шлях_до_програми> (<аргументи>) (<тип даних>:<результат>)

Наприклад:

c:\mydir\gcd.py (5,10) (int:5)

31.21. Скласти програму у якій основний потік запускає новий потік і далі нічого не робить. Новий потік рахує час і показує час (години, хвилини) кожную нову хвилину.

Проекти для розробки

1. Інтернет-крамниця

Скласти програму, що працює в оточенні веб-сервера та реалізує просту інтернет-крамницю. Товари у крамниці розбиті на категорії. Для кожного товару доступна назва, опис, характеристики, фото.

Дані про товари зберігаються у базі даних.

Програма повинна здійснювати пошук за категорією товарів, за частиною назви товару, показувати знайдений товар (товари) дозволяти вибрати товар, вказати кількість товару, додати товар до корзини/видалити раніше доданий товар, та показувати сторінку з вибраними товарами та їх загальною вартістю.

2. Склад (веб)

Скласти програму, що працює в оточенні веб-сервера та реалізує функціональність роботи складу. На склад надходять та зберігаються партії товару. Товари діляться по категоріях. Кожна партія зберігається у виділеному місці (корпус, відділення, полиця).

Інформація про товари зберігається у базі даних.

Програма повинна забезпечити пошук за категорією товару, за частиною назви товару, показати залишок певного товару на складі та місце зберігання для кожної партії вибраного товару. Також програма повинна підтримувати приймання товарів на склад (введення інформації про товар та його зберігання) а також відпуск товару зі складу (вказання місця, з якого забирають товар, виведення накладної на товар (товари) за заданим шаблоном накладної у MS Word.

3. Склад (локальний)

Скласти програму з графічним інтерфейсом, що реалізує функціональність роботи складу. На склад надходять та зберігаються партії товару. Товари діляться по категоріях. Кожна партія зберігається у виділеному місці (корпус, відділення, полиця).

Інформація про товари зберігається у базі даних.

Програма повинна забезпечити пошук за категорією товару, за частиною назви товару, показати залишок певного товару на складі та місце зберігання для кожної партії вибраного товару. Також програма повинна підтримувати приймання товарів на склад (введення інформації про товар та його зберігання) а також відпуск товару зі складу (вказання місця, з якого забирають товар, виведення накладної на товар (товари) за заданим шаблоном накладної у MS Word.

4. Торгівельна компанія

Скласти програму з графічним інтерфейсом, що реалізує функціональність роботи торговельної компанії. Компанія здійснює гуртову торгівлю товарами декількох категорій. У компанії є ряд постачальників та клієнтів.

Інформація про товари, постачальників та клієнтів зберігається у базі даних.

Програма повинна дозволяти: додати/змінити/видалити постачальника, додати/змінити/видалити клієнта, додати/змінити/видалити товар, прив'язати/відв'язати товар до/від постачальника, сформуванню замовлення для постачальника, зафіксувати отримання товарів, створити рахунок для клієнта (клієнт замовляє товари телефоном).

5. Гра у шахи у мережі

Скласти програму, що працює в оточенні веб-сервера а також клієнта з графічним інтерфейсом для реалізації гри двох гравців у шахи у мережі. Програма повинна підтримувати підключення двох гравців, починати та вести гру, передаючи ходи одного гравця іншому, зберігати партії у базі даних, надавати можливість «програти» раніше зіграні партії по кроках.

6. Віртуальний «диск» у мережі

Скласти програму, яка працює в оточенні веб-сервера а також клієнта з графічним інтерфейсом, які підтримують віртуальний «диск» у мережі. Користувачі після реєстрації можуть зберегти свої файли у структурі каталогів на сервері, створити новий каталог або підкаталог, завантажити раніше збережені файли з сервера, подивитись зміст каталогу(ів), вибрати файли для завантаження.

7. Домашній бюджет (веб)

Скласти програму, яка працює в оточенні веб-сервера та відслідковує домашній бюджет. Програма повинна забезпечити фіксацію надходження (введення) коштів з різних джерел а також фіксацію їх розподілення по статтях: їжа, комунальні витрати, транспорт, розваги тощо. Для кожного платежу також вводиться його деталізація (компанія для комунальних витрат, інше).

Дані повинні зберігатись у базі даних.

Програма повинна також забезпечити пошук надходжень та витрат за вибраний період, відображення статистики витрат по категоріях, компаніях, показ графіків (Excel або matplotlib) витрат за період по категоріях.

8. Домашній бюджет (локальний)

Скласти програму з графічним інтерфейсом, яка відслідковує домашній бюджет. Програма повинна забезпечити фіксацію надходження (введення) коштів з різних джерел а також фіксацію їх розподілення по статтях: їжа, комунальні витрати, транспорт, розваги тощо. Для кожного платежу також вводиться його деталізація (компанія для комунальних витрат, інше).

Дані повинні зберігатись у базі даних.

Програма повинна також забезпечити пошук надходжень та витрат за вибраний період, відображення статистики витрат по категоріях, компаніях, показ графіків (Excel або matplotlib) витрат за період по категоріях.

9. Інтернет-моніторинг

Скласти програму з графічним інтерфейсом, яка здійснює моніторинг Інтернет щодо заданої тематики. Тематика визначається набором ключових слів. Також задається перелік сайтів, на яких здійснюється моніторинг.

Тематики та перелік сайтів зберігається у базі даних. Також у базі даних зберігаються результати моніторингу: статті та посилання на них.

Програма повинна складатись з двох частин: графічний інтерфейс для додавання/зміни/видалення тематики, додавання/зміни/видалення сайту, показу результатів моніторингу за період; а також сам клієнт моніторингу, що запускається за графіком, збирає інформацію та зберігає її у базі даних.

10. Планування інтер'єру

Скласти програму з графічним інтерфейсом для реалізації планування інтер'єру помешкання. Програма повинна дозволити додати/змінити/видалити кімнату заданих розмірів додати/змінити/видалити меблі, подивитись спланований інтер'єр, вибравши кімнату. Кожен елемент меблів зображувати прямокутником заданих розмірів та назвою. Меблі можна пересувати та повертати на заданий кут. Меблі не повинні перетинатися після переміщення. Кімнати разом з меблями а також перелік меблів зберігаються у базі даних.

11. Нарахування зарплати (веб)

Скласти програму, яка працює в оточенні веб-сервера та здійснює нарахування заробітної платні співробітникам підприємства. Нарахування здійснюється за табелем, що заповнюється на місяць. У таблиці за кожен день місяця вказують кількість відпрацьованих годин або «в» - відпустка або «л» - лікарняний. За кожний місяць відома загальна кількість годин, яку треба відпрацювати. Понаднормові години не враховуються. Заробітна платня визначається посадою, яку обіймає співробітник та кількістю відпрацьованих годин. Робітники у відпустці отримують заробітну платню за день у розмірі середнього на день значення отриманої платні за останній рік. За лікарняний – 80% заробітної платні, що визначається посадою.

Дані робітників, таблиці, відомості про нарахування зберігати у базі даних.

Програма повинна дозволити додати/змінити/видалити дані робітника, додати/змінити/видалити дані посади, додати/змінити/видалити таблиць робітника за вказаний місяць, показати дані про нарахування усім робітникам за місяць або вибраному робітнику за період.

12. Нарахування зарплати (локальне)

Скласти програму з графічним інтерфейсом, яка здійснює нарахування заробітної платні співробітникам підприємства. Нарахування здійснюється за табелем, що заповнюється на місяць. У таблиці за кожен день місяця вказують кількість відпрацьованих годин або «в» - відпустка або «л» - лікарняний. За кожний місяць відома загальна кількість годин, яку треба

відпрацювати. Понаднормові години не враховуються. Заробітна платня визначається посадою, яку обіймає співробітник та кількістю відпрацьованих годин. Робітники у відпустці отримують заробітну платню за день у розмірі середнього на день значення отриманої платні за останній рік. За лікарняний – 80% заробітної платні, що визначається посадою.

Дані робітників, таблиці, відомості про нарахування зберігати у базі даних.

Програма повинна дозволяти додати/змінити/видалити дані робітника, додати/змінити/видалити дані посади, додати/змінити/видалити таблиць робітника за вказаний місяць, показати дані про нарахування усім робітникам за місяць або вибраному робітнику за період.

13 Кадри (веб)

Скласти програму, яка працює в оточенні веб-сервера та реалізує просту систему обліку кадрів підприємства. Забезпечити додавання/зміну/видалення інформації про робітника, додавання/зміну видалення інформації про підрозділи підприємства, посади, кількість штатних одиниць по підрозділах. Також програма повинна будувати накази про прийняття робітника на роботу, переведення на посаду (у підрозділ), звільнення у XML.

Інформація про робітників, посади, підрозділи, накази зберігається у базі даних.

14 Кадри (локальні)

Скласти програму з графічним інтерфейсом, яка реалізує просту систему обліку кадрів підприємства. Забезпечити додавання/зміну/видалення інформації про робітника, додавання/зміну видалення інформації про підрозділи підприємства, посади, кількість штатних одиниць по підрозділах. Також програма повинна будувати накази про прийняття робітника на роботу, переведення на посаду (у підрозділ), звільнення за шаблонами у MS Word.

Інформація про робітників, посади, підрозділи, накази зберігається у базі даних.

15. Торгівля автозапчастинами (веб)

Скласти програму, яка працює в оточенні веб-сервера та реалізує функціональність торгівлі автозапчастинами. Автозапчастини визначаються маркою, моделлю та роком випуску автомобіля, належать до певної категорії, мають назву та код. Програма повинна забезпечити додавання/зміну/видалення автомобіля, додавання/зміну/видалення запчастини автомобіля, пошук запчастин за назвою або категорією, приймання замовлення від клієнта на автозапчастини та формування специфікації з переліком запчастин та цін у форматі JSON.

Дані про автомобілі, запчастини, замовлення зберігаються у базі даних.

16. Торгівля автозапчастинами (локальна)

Скласти програму з графічним інтерфейсом, яка реалізує функціональність торгівлі автозапчастинами. Автозапчастини визначаються маркою, моделлю та роком випуску автомобіля, належать до певної категорії, мають назву та код. Програма повинна забезпечити додавання/зміну/видалення автомобіля, додавання/зміну/видалення запчастини автомобіля, пошук запчастин за назвою або категорією, введення замовлення на автозапчастини, отримане від клієнта, та формування специфікації з переліком запчастин та цін у форматі MS Excel.

Дані про автомобілі, запчастини, замовлення зберігаються у базі даних.

17. Домашня бібліотека

Скласти програму з графічним інтерфейсом, яка реалізує функціональність, що необхідна для ведення домашньої бібліотеки. У бібліотеці зберігаються книги та періодичні видання. Усі видання розподілені за жанрами (може бути декілька для одного видання), видом видання (науково-технічне, підручник, художня література), рік випуску. У кожного видання може бути один або декілька авторів. Так само вказують інформацію щодо місця збереження видання у бібліотеці (шафа, полиця).

Програма повинна дозволити додавання/зміну/видалення видання, додавання/зміну/видалення автора додавання/зміну/видалення виду або жанру, місць збереження.

Також програма повинна здійснювати пошук видання (видань) за назвою, автором, жанром, видом.

Дані про видання, авторів, види, жанри, місця збереження зберігаються у базі даних.

Список джерел

Задачники з програмування

1. Збірник задач з дисципліни "Інформатика і програмування" / Є.С. Вакал, В.В. Личман, О.В. Обвінцев, В.В. Бублик, Б.П. Довгий, В.В. Попов. –2-ге вид., випр. та доп. –К.: ВПЦ "Київський університет", 2006.– 94 с.
2. Крєневич А.П., Обвінцев О.В. С у задачах і прикладах : навчальний посібник із дисципліни "Інформатика та програмування" – К.: ВПЦ "Київський університет", 2012. – 211 с.
3. Абрамов, С.А. Задачи по программированию / С.А. Абрамов, Г.Г. Гнездилова, Е.Н. Капустина, М.И. Селюн. – М.: Наука, 1988. – 224 с

Книги до курсів

4. Обвінцев О.В. Інформатика та програмування. Курс на основі Python. Матеріали лекцій. – К., Основа, 2017
5. Обвінцев О.В. Об'єктно-орієнтоване програмування. Курс на основі Python. Матеріали лекцій. – К., Основа, 2017
6. Крєневич А. П. Python у прикладах і задачах. Частина 1. Структурне програмування Навчальний посібник із дисципліни "Інформатика та програмування" [Електронний ресурс] / Андрій Павлович Крєневич. – 2017. – Режим доступу до ресурсу:

- <http://www.matfiz.univ.kiev.ua/uploads/books/python2017.pdf>
7. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування, <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/01/python-u-prykladakh-i-zadachakh.-ch2-oor.pdf>
 8. Крєневич А. П. Методичні вказівки до лабораторних занять із дисципліни «Програмування» для студентів механіко-математичного факультету [Електронний ресурс] / Андрій Павлович Крєневич. – 2020. – Режим доступу до ресурсу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/01/metodychni-vkazivky-prohramuвання.pdf>
 9. Крєневич А.П. Алгоритми і структури даних. Підручник. – К.: 2021. – 200 с. – Режим доступу до ресурсу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/09/pidruchnyk-alhorytmy-i-struktury-danykh.pdf>
 10. Бородін В.А. Методичні вказівки з дисципліни “Мова програмування С++” Методична розробка. 101 с. [Електронний ресурс]. – Режим доступу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/11/method.pdf>
 11. Гап'як І.В., Довгий Б.П., Вакал Є.С., Ловейкін А.В. Методичні вказівки до застосування математичних пакетів для чисельно-аналітичного розв'язування задач. http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/04/hapiak_dovhyj_vakal_lovejkin_metodychni-vkazivky.pdf
 12. A Byte of Python (Russian) Версия 2.01 Swaroop С Н (Translated by Vladimir Smolyar),

<http://wombat.org.ua/AByteOfPython/AByteofPythonRussia-n-2.01.pdf>

13. Марк Лутц, Изучаем Python, 4-е издание, 2010, Символ-Плюс
14. Python 3.4.3 documentation.
<https://docs.python.org/3.4/index.html>
15. Бублик В.В., Личман В.В., Обвінцев О.В.. Інформатика та програмування. Електронний конспект лекцій, 2003 р.,
<http://www.matfiz.univ.kiev.ua/books> (також на
<http://obvintsev.info/compuscience/lectures/index.htm>)
16. Самоучитель Python. <http://pythonworld.ru/samouchitel-python>
17. С. Шапошникова. Основы программирования на Python. Версия 2 (2011). <http://younglinux.info/pdf>
18. Бизли Д. Python. Подробный справочник. - Символ-Плюс, 2010.
19. Марк Саммерфилд, Программирование на Python 3. Подробное руководство. - Символ-Плюс, 2009.
20. Doug Hellmann. Python Module of the Week. - <http://pymotw.com/2/#>
21. Matt Telles. Python Power! The Comprehensive Guide. - Thomson Course Technology, 2012.
22. Mark Lutz. Programming Python 4th Edition. - O'Reilly Media, 2010.
23. Tarek Ziadé. Expert Python Programming. - Packt Publishing, 2008.
24. Марк Саммерфилд - Python на практике. ДМК - 2014
25. Langtangen H.P. - A Primer on Scientific Programming with Python, 2nd Edition – 2011
26. Wesley J. Chun - Core Python Programming - 2001
27. Peter Norton, Alex Samuel, David Aitel та інші - Beginning Python

28. Magnus Lie Hetland - Beginning Python from Novice to Professional, 2nd ed – 2008
29. Mark Pilgrim - Dive into Python, Version 5.4 - 2004
30. Noah Gift, Jeremy M. Jones - Python for Unix and Linux System Administration
31. John Goerzen -Foundations of Python Network Programming. - 2004
32. Paul Gries and Others - Practical Programming - An Introduction to Computer Science Using The Python 3 - 2nd Edition – 2013
33. Bernd Klein. Python Course <http://www.python-course.eu/index.php>

Книги з програмування

34. Вирт Н. Алгоритмы + структуры данных = программы.- М.: Мир, 1985.- 406с.
35. Грис Д. Наука программирования .-М.: Мир, 1984.- 416с.
36. Кнут Д.Э. Искусство программирования (в 3 томах). – М.: Издательский дом «Вильямс», 2002

Україномовні ресурси з Python

37. Марк Пілігрим. Пориньте у Python 3.
https://uk.wikibooks.org/wiki/%D0%9F%D0%BE%D1%80%D0%B8%D0%BD%D1%8C%D1%82%D0%B5_%D1%83_Python_3
38. Програмування на мові Python (3.x). Початковий курс.
<https://sites.google.com/site/pythonukr/>
39. Віталій Подоба. Python 2: Курс Молодого Бійця.
<http://www.vitaliypodoba.com/tutorials/python2-beginners-course/>
40. Віталій Подоба. Веб-розробка з Python та Django для початківців. <http://www.vitaliypodoba.com/books/django-for-beginners/>