

Міністерство освіти і науки України
Сумський державний педагогічний університет імені
А.С.Макаренка

В.М.Базурін
О.С.Чашечникова

Мова програмування Python. Початковий рівень
Збірник задач

Суми 2021

Б17

Рекомендовано до друку Вченою радою Сумського державного педагогічного університету імені А.С.Макаренка (протокол №2 від 27.09.2021 р.)

Рецензенти:

Семеніхіна О.В. – доктор педагогічних наук, професор, завідувач кафедри інформатики Сумського державного педагогічного університету імені А.С.Макаренка

Рябко А.В. – кандидат педагогічних наук, старший викладач кафедри фізико-математичної та інформатичної освіти та комп'ютерних технологій Глухівського національного педагогічного університету імені Олександра Довженка

Б17	Базурін В.М., Чашечникова О.С. Мова програмування Python. Початковий рівень. Збірник задач. Суми: ФОП Цьома С.П., 2022. – 176 с.
	У збірнику задач наведено задачі з програмування, які враховують особливості мови Python. Збірник задач розраховано на учнів і студентів, які починають вивчення мови програмування Python

Зміст

Вступ.....	4
Розділ 1. Алгоритмічна констукція «Слідування».....	5
Розділ 2. Алгоритмічна структура «Розгалуження».....	23
Розділ 3. Алгоритмічна структура «Цикл».....	33
Розділ 4. Комп'ютерна графіка в Python	48
Розділ 5. Алгоритми обробки рядкових і символьних величин	60
Розділ 6. Одновимірні масиви	70
Розділ 7. Двовимірні масиви.....	86
Розділ 8. Списки і кортежі	105
Розділ 9. Словники	121
Розділ 10. Підпрограми	131
Розділ 11. Модулі.....	152
Розділ 12. Файли	161
Список використаних і рекомендованих джерел.....	175
Додатки.....	176

Вступ

У даний час професії, пов'язані з галуззю ІТ, є одними з найкраще оплачуваних і найбільш востребуваних на ринку праці. Для того щоб вивчитись за однією з таких професій, слід довго і навчатись.

Значна кількість абітурієнтів, які вступили у вищі заклади освіти на ІТ-спеціальності, не розробили жодної програми і лише уявляють, як це робити. Тому вони виявляються невідготовленими до вивчення програмування у вищих закладах освіти, де навчання програмування проходить досить інтенсивно: кожна лабораторна робота є досить складною і, щоб успішно виконати її, слід мати базові навички з програмування.

Для того, щоб успішно вивчати програмування у вищих закладах освіти, абітурієнт повинен мати навички створення програм. У шкільному курсі інформатики основною мовою програмування є Python. Відповідно навчати учнів створювати програми на мові Python повинен вчитель, який сам добре володіє цією мовою.

Мова Python є потужним засобом програмування, який широко застосовується і в розробці серверних додатків, прикладних програм, комп'ютерних ігор, програмування математичних та інженерних розрахунків, розв'язування задач оптимізації тощо. Тому вивчення мови Python може стати у нагоді як майбутнім учителям інформатики, так і майбутнім інженерам-будівельникам, фахівцям галузі легкої промисловості тощо. Головна цінність мови Python у тому, що для неї розроблена значна кількість модулів, які розширюють і поглиблюють можливості цієї мови. Більшість цих модулів поширюються вільно і безкоштовно, тому їх використання є цілком виправданим.

Пропонований збірник задач розрахований на початківця, який тільки починає програмувати на мові Python. Задачі мають нескладний алгоритм розв'язання і є цілком посильними як для учнів 7-8 класів, так і для студентів вищих закладів освіти. Посібник побудовано з принципом «від простого до складного», тобто спочатку треба розв'язати простіші задачі, а потім – складніші.

Посібник налічує 12 розділів, які охоплюють алгоритмічні конструкції і основні типи даних, які використовуються у мові Python. Для написання програм не потрібно встановлювати додаткові бібліотеки (крім розділу «Двовимірні масиви»), де для створення програм використовується бібліотека NumPy).

Розділ 1. Алгоритмічна констукція «Слідування»

Основні арифметичні дії на мові Python наведено в табл.1.1.

Таблиця 1.1

Основні арифметичні дії

Позначення	Назва
+	Додавання
-	Віднімання
*	Множення
/	Ділення
//	Ділення націло
%	Остача від ділення

Під час ділення наприклад $1/x$ слід ставити крапку після 1 (1.) для того, щоб відбулося не ділення націло, а ділення взагалі.

Модулі

Підключення модулів на мові Python

Стандартні команди мови Python доповнюються додатковими командами, зміст яких прописано в модулях. Модулі мають різну назву у залежності від тих команд, які в них містяться. Для підключення модуля до програми достатньо ввести на початку програми команду:

```
import <назва модуля>
```

Наприклад, для підключення модуля, який містить математичні функції, треба набрати команду:

```
import math
```

Для того, щоб підключити модуль Черепаха, необхідно ввести команду:

```
import turtle
```

Розглянемо деякі модулі, які є стандартними.

Модуль `math` містить кілька функцій, опис яких наведено в табл.1.2.

Таблиця 1.2

Функції модуля `math`

Назва функції	Опис	Приклад використання
<code>math.fabs(x)</code>	Модуль числа x	<code>y=math.fabs(12)</code>
<code>math.ceil(x)</code>	Округлення до найближчого більшого числа	<code>y=math.ceil(2.7)</code>
<code>math.factorial(x)</code>	Факторіал числа x	<code>y=math.factorial(4)</code>

<code>math.sqrt(x)</code>	Квадратний корінь із числа x	<code>y=math.sqrt(4)</code>
<code>math.pow(x,y)</code>	Піднесення числа x до степеня y	<code>y= math.pow(2,4)</code>

Модуль `random` містить функції, які використовуються для роботи з випадковими числами.

Таблиця 1.2

Функції модуля `random`

Назва функції	Опис	Приклад використання
<code>random.randint(a,b)</code>	Випадкове ціле число у межах від a до b	<code>z=random.randint(3,25)</code>
<code>random.random()</code>	Випадкове дійсне число в межах від 0 до 1	<code>q=random.random()</code>
<code>random.choice(['a','o','i','e','u'])</code>	Випадковий символ із списку	<code>random.choice(['a','o','i','e','u'])</code>

Приклади задач

Приклад 1.1. Написати програму для обчислення периметра і площі прямокутника, заданого своїми сторонами.

Текст програми (linear02)

```
a=float(input("Введіть a: "))
b=float(input("Введіть b: "))
S=a*b
p=2*(a+b)
print("S=",S)
print("P=",p)
```

Приклад 1.2. Написати програму для обчислення площі поверхні та об'єму куба.

Текст програми (linear03)

```
a=float(input("Введіть a: "))
S=6*a*a
V=a*a*a
print("S=",S)
print("V=",V)
```

Якщо використати імпорт функцій, то програма змінить вигляд:

Текст програми (linear03a)

```
from math import pow
a=float(input("Введіть a: "))
S=6*pow(a,2)
V=pow(a,3)
```

```
print("S=",S)
print("V=",V)
```

Приклад 1.3. Написати програму для обчислення гіпотенузи прямокутного трикутника, якщо його катети задано користувачем.

Текст програми (linear01)

```
from math import sqrt
a=float(input("Введіть сторону a"))
b=float(input("Введіть сторону b"))
c=sqrt(a**2+b**2)
print("c=",c)
```

Задачі для виконання

1.1. Дано радіус кола. Написати програму, яка обчислює його площу і довжину кола.

1.2. Користувач вводить масу і об'єм тіла. Програма повинна обчислити густину речовини.

1.3. Написати програму, яка обчислює площу кільця. Користувач вводить діаметри внутрішнього і зовнішнього кіл.

1.4. Написати програму, яка обчислює середнє арифметичне трьох чисел, введених з клавіатури.

1.5. Написати програму, яка обчислює периметр прямокутника і його діагональ. Користувач вводить сторони прямокутника.

1.6. Користувач вводить сторони трикутника. Написати програму, яка обчислює площу і периметр трикутника.

1.7. У програмі задано ціну 1 кг цукерок, 1 кг печива і 1 кг апельсинів. Написати програму, яка обчислить вартість покупки x_1 кг цукерок, x_2 кг печива, x_3 кг апельсинів. Користувач повинен вводити лише значення x_1 , x_2 та x_3 .

1.8. Написати програму, яка обчислює довжину дуги, що дорівнює $5/9$ довжини кола радіусом R . Користувач вводить значення R .

1.9. Написати програму, яка перераховує масу з кілограмів у фунти.

1.10. Написати програму, яка перераховує довжину з метрів у фути.

1.11. Прямокутник розміщено так, що його центр співпадає з центром координат, а його сторони паралельні осям координат. Користувач задає координати лівої верхньої і правої нижньої вершин прямокутника. Написати програму, яка виводить у два рядки координати x та y двох інших вершин прямокутника.

1.12. У Західній Європі використовується міра об'єму рідини 1 баррель=158.987295 л. Написати програму, яка перераховує об'єм з кубічних метрів у баррелі.

1.13. У Західній Європі використовується міра об'єму рідини 1 галлон=3,785 л. Написати програму, яка перераховує об'єм з літрів у галлони.

1.14. У суднобудуванні використовується міра місткості 1 брутто-тонна(брт)=2,83 м³. Написати програму для перерахунку об'єму з кубічних метрів у брутто-тонни.

1.15. Шкільне подвір'я має форму прямокутника розмірами ахв м і оточено парканом висотою 1,1 м. Для пофарбування паркану треба закупити фарбу в банках по 2,8 кг. Програма повинна визначити кількість банок фарби, які потрібно купити, якщо на 1 м² паркану витрачається 220 г фарби.

1.16. Інформація вимірюється в одиницях байт, 1 кілобайт (КБ) =1024 КБ, 1 мегабайт (МБ) = 1024 КБ. Написати програму, яка перераховує кілобайти у байти.

1.17. Інформація вимірюється в одиницях байт, 1 кілобайт (КБ) =1024 КБ, 1 мегабайт (МБ) = 1024 КБ. Написати програму, яка перераховує мегабайти у байти.

1.18. Стальний бак має форму циліндра з діаметром основи d і висотою h. Написати програму, яка обчислює об'єм бака.

1.19. Швидкість руху автомобіля 60 км/год. Написати програму, яка підраховує пройдений автомобілем шлях за час t. Величина t вводиться з клавіатури.

1.20. Автомобіль рухався час t1 з швидкістю v1, а час t2 – з швидкістю v2. Написати програму, яка обчислить середню швидкість автомобіля.

1.21. Алюмінієвий бак має форму прямокутного паралелепіпеда (без кришки) довжиною а м, шириною b м, висотою с м. Написати програму, яка обчислює площу поверхні бака.

1.22. Зустріч з науковцем тривала х хв. Написати програму, яка перетворить хвилини у години і хвилини.

1.23. Поле має форму паралелограма з сторонами а і b (м). Урожайність пшениці на ньому становила 25 ц/га. Написати програму, яка обчислить урожай в тоннах, зібраний фермером з поля.

1.24. Борошно розфасовано в мішки по 50 кг. Вантажопідйомність фургона становить х т. Написати програму, яка обчислить кількість мішків борошна, які можна перевезти у фургоні.

1.25. Ділянка землі має форму опуклого чотирикутника з довжиною сторін a1, a2, a3, a4 (м). Висота огорожі h (м). Написати програму, яка обчислить площу огорожі.

1.26. Покупка складається з n_1 зошитів, n_2 книжок і n_3 ручок. Ціна цих товарів різна. Написати програму, яка обчислює вартість покупки.

1.27. Користувач вводить двоцифрове число. Програма повинна вивести в окремих рядках першу і другу цифру.

1.28. Користувач вводить трицифрове число. Програма повинна вивести у стовпчик всі цифри.

1.29. Користувач вводить двоцифрове число. Програма повинна обчислити суму цифр числа.

1.30. Користувач вводить трицифрове число. Програма повинна обчислити суму цифр числа.

1.31. Користувач вводить чотирицифрове число. Програма повинна обчислити суму всіх цифр.

1.32. Користувач задає інтервал часу в секундах. Програма повинна вивести цей час в годинах, хвилинах і секундах.

1.33. Написати програму, яка обчислює суму двох випадкових чисел: перше в межах $[0; 50]$, друге – в межах проміжку $[60; 100]$.

1.34. Написати програму, яка обчислює діагональ куба. Користувач вводить довжину ребра.

1.35. Написати програму, яка обчислює сторону ромба. Користувач вводить довжини діагоналей.

1.35. Користувач вводить довжини діагоналей ромба. Програма повинна обчислити периметр ромба.

1.36. Написати програму, яка обчислить відстань від точки до центра координат. Користувач вводить координати точки $(x; y)$.

1.37. Написати програму, яка обчислюватиме відстань між двома точками. Користувач вводить координати двох точок.

1.38. Користувач вводить довжину гіпотенузи і одного з катетів прямокутного трикутника. Програма повинна обчислити довжину другого катета.

1.39. Користувач вводить величину гострого кута прямокутного трикутника. Програма повинна обчислити величину другого гострого кута.

1.40. Написати програму, яка перераховує відстань з футів в метри ($1 \text{ фут} = 30,3 \text{ см}$).

1.41. Написати програму, яка перераховує розміри з дюймів у сантиметри ($1 \text{ дюйм} = 2,5 \text{ см}$).

1.42. Написати програму, яка обчислює середнє арифметичне двох чисел, введених користувачем.

1.43. Написати програму, яка обчислює середнє арифметичне трьох чисел, введених користувачем.

1.44. Користувач вводить число. Програма повинна додати до нього випадкове число в межах від 50 до 80 і вивести результат на екран.

1.45. Користувач вводить два числа. Програма повинна обчислити суму їх модулів.

1.46. Користувач вводить 3 числа. Програма повинна обчислити суму їх квадратів.

1.47. Користувач вводить 2 числа. Програма повинна обчислити суму їх кубів.

1.48. Користувач вводить 2 числа. Програма повинна обчислити різницю їх квадратів.

1.49. Написати програму, яка обчислить середнє геометричне двох чисел.

1.50. Написати програму, яка обчислить середнє геометричне трьох чисел.

1.51. Користувач вводить два числа, більших від нуля. Програма повинна обчислити суму квадратних коренів цих чисел.

1.52. Користувач вводить два числа, більших від нуля. Програма повинна обчислити добуток квадратних коренів цих чисел.

1.53. Користувач вводить швидкість передачі даних через оптоволоконну лінію у КБ/с. Програма повинна обчислити час, необхідний для передачі фільму (файлів у форматі DVD обсягом 4,7 ГБ).

1.54. Написати програму, яка обчислює тиск стовпа ртуті висотою h мм. Густина ртуті становить $13,3 \text{ г/см}^3$.

1.55. Написати програму, яка обчислює тиск стовпа води висотою h м. Густина води 1000 кг/м^3 .

1.56. Написати програму, яка обчислює вагу кубика з довжиною ребра x см, виготовленого з сталі (густина сталі – $7,8 \text{ г/см}^3$).

1.57. Написати програму, яка обчислює вагу мідної кульки діаметром 6 см. Густина міді – $8,3 \text{ г/см}^3$.

1.58. Користувач вводить довжини сторін двох прямокутників. Програма повинна обчислити суму діагоналей обох прямокутників.

1.59. Написати програму, яка обчислює довжину ламаної лінії, яка складається з 4 ланок. Користувач вводить довжини цих ланок.

1.60. Користувач вводить довжини сторін трикутника. Програма повинна обчислити площу трикутника.

1.61. Користувач вводить дві сторони паралелограма і кут між ними. Програма повинна обчислити площу паралелограма.

1.62. Користувач вводить довжини основ трапеції і її висоту. Програма повинна обчислити площу трапеції.

1.63. Написати програму, яка обчислить третій кут трикутника, якщо користувач ввів два інших кута.

1.64. Написати програму, яка обчислить синус і тангенс кута. Користувач вводить косинус кута.

1.65. Користувач вводить радіуси двох кіл. Програма повинна обчислити суму площ обох кіл.

1.66. Написати програму, яка обчислить сумарний об'єм двох прямокутних паралелепіпедів. Розміри паралелепіпедів вводяться з клавіатури.

1.67. Написати програму, яка обчислює тангенс і котангенс кута трикутника, якщо користувач вводить довжини його катетів.

1.68. Написати програму, яка обчислює площу рівностороннього трикутника за його трьома сторонами.

1.69. Написати програму, яка обчислює густину цегли, якщо користувач задає масу цеглини і її розміри.

1.70. Жорсткий диск комп'ютера має обсяг x ГБ. Написати програму, яка обчислить кількість фільмів у форматі BlueRays (1 фільм=32 ГБ), які помістяться на диску.

1.71. Один DVD диск подвійної щільності може містити 9,4 ГБ даних. Написати програму, яка визначить, скільки таких дисків поміститься на жорсткому диску комп'ютера. Обсяг жорсткого диску вводить користувач.

1.72. Один DVD диск може містити 8,5 ГБ даних. Визначити, скільки таких дисків поміститься на жорсткому диску комп'ютера. Обсяг жорсткого диску в терабайтах вводить користувач.

1.73. Один CD-диск може вміщувати від 650 МБ до 700 МБ даних. Написати програму, яка обчислить, на скільки таких дисків слід зберегти дані з вінчестера обсягом x ГБ під час резервного копіювання.

1.74. Хлопчику потрібно перенести з комп'ютера на комп'ютер багато фільмів по z МБ у кожному. Для цього хлопчик може використати флеш-карту обсягом 16 ГБ. Написати програму, яка обчислить кількість фільмів, які можна перенести за один раз за допомогою флеш-карти.

1.75. Під час резервного копіювання даних виникла потреба зберегти їх на іншому комп'ютері. Хлопчик перевірів обсяг даних на жорсткому диску, які необхідно перенести, і виявив, що їх q (ГБ). У хлопчика є дві флеш-карти – 8 ГБ і 16 ГБ. Написати програму, яка визначить мінімальну кількість перенесень файлів, які треба виконати хлопчику.

1.76. Текст містить X символів. Кожна сторінка містить 32 рядки по 28 символів у кожному. Написати програму, яка визначить кількість сторінок, які займає текст.

1.77. Книга містить A сторінок по B рядків на кожній. Кожен рядок налічує C символів. Кожен символ кодується 1 байтом. Написати програму, яка визначить обсяг пам'яті, потрібний для зберігання цієї книги. Величини A , B , C вводяться з клавіатури.

1.78. Користувач вводить радіуси трьох кіл. Написати програму, яка обчислить суму довжин всіх трьох кіл.

1.79. Користувач вводить діаметри і висоти двох циліндрів. Написати програму, яка обчислить сумарний об'єм обох циліндрів.

1.80. Бак діаметром D м і висотою h м на $2/3$ заповнено бензином. Написати програму, яка обчислить масу бензину в баку, якщо його густина 7800 кг/м^3 .

1.81. Гас наливо в бак, який має форму прямокутного паралелепіпеда, в основі якого лежить квадрат зі стороною q (м). Висота бака h (м). густина гасу 6700 кг/м^3 . Програма повинна обчислити масу гасу в баку в тоннах.

1.82. Олію наливо в циліндричний бак радіусом R і висотою h (м). Густина олії знайти самостійно. Написати програму, яка обчислить масу олії в баку.

1.83. Циліндричне відро має діаметр основи 30 см, а висоту – 40 см. Порожнє відро важить 1,7 кг. Написати програму, яка обчислить масу повного відра з рідиною. Густина рідини вводить користувач.

1.84. Свинцевий циліндр має такі розміри: діаметр D (см) і висоту h (см). Густина свинцю $9,3 \text{ г/см}^3$. Написати програму, яка обчислює масу і вагу циліндра.

1.85. Мідна кулька радіусом R (см) лежить на столі. Написати програму, яка обчислить масу кулі і силу, з якою куля діє на опору.

1.86. Користувач вводить опори трьох резисторів, встановлених послідовно. Програма повинна обчислити загальний опір цих резисторів.

1.87. Користувач вводить опори трьох резисторів, встановлених паралельно. Програма повинна обчислити загальний опір цих резисторів.

1.88. Написати програму для обчислення потенціальної енергії тіла. Маса тіла і висоту вводять з клавіатури.

1.89. Користувач вводить два числа. Програма повинна обчислити остачу від ділення першого числа на 3, остачу від ділення другого числа на 8, а потім вивести суму цих остач.

1.90. Написати програму для обчислення значення функції $f=3x^4-x^2$.

1.91. Написати програму для обчислення значення функції $y=x^3+12x+3$.

1.92. Написати програму, яка обчислює значення функції $y=14-x^4$.

1.93. Коло радіусом R поділено на N однакових секторів. Написати програму, яка обчислює площу одного сектора. Користувач вводить R і N .

1.94. Написати програму, яка обчислює площу поверхні кулі. Радіус кулі вводиться з клавіатури.

1.95. Основою піраміди є прямокутник зі сторонами a і b . Висота піраміди h . Написати програму для обчислення об'єму піраміди.

1.96. Користувач вводить радіус основи конуса і його висоту. Програма повинна обчислити об'єм конуса.

1.97. Написати програму, яка обчислює різницю коренів двох додатних дійсних чисел, введених користувачем.

1.98. Написати програму, яка обчислює суму двох векторів, заданих координатами.

1.99. Написати програму, яка обчислює скалярний добуток двох векторів. Користувач вводить модулі векторів і кут між ними.

1.100. Написати програму, яка обчислює скалярний добуток двох векторів за їх координатами на площині.

Цілочисельна арифметика

Приклад 1.4. Написати програму, яка обчислює, скільки цілих метрів у відрізку, що дорівнює Z см. Користувач вводить величину Z .

Текст програми (linear04)

```
sm=int(input("Введіть довжину у сантиметрах: "))
Z=int(sm/100)
print("у цьому числі цілих метрів: ",Z)
```

Приклад 1.5. Написати програму, яка обчислює кількість цілих квадратів зі стороною x , які може містити прямокутник розмірами $a \times b$. Користувач вводить числа x , a , b .

Текст програми (linear05)

```
a=int(input("Введіть a"))
b=int(input("Введіть b"))
x=int(input("Введіть x"))
k1=int(a/x)
k2=int(b/x)
k=k1*k2
print("у прямокутнику міститься ",k," квадрати")
```

Задачі для розв'язування

1.101. Користувач вводить двоцифрове число. Написати програму, яка виведе на екран дві цифри окремо.

1.102. Користувач вводить трицифрове число. Написати програму, яка виведе на екран три цифри окремо.

1.103. Користувач вводить двоцифрове число. Написати програму, яка виведе на екран суму цифр числа.

1.104. Користувач вводить трицифрове число. Написати програму, яка виведе на екран суму цифр числа.

1.105. Користувач вводить трицифрове число. Написати програму, яка вилучить з числа середню цифру. Роботу з рядковими величинами не використовувати.

1.106. Користувач вводить чотирицифрове число. Написати програму, яка обчислює суму цифр числа.

1.107. Користувач вводить чотирицифрове число. Написати програму, яка відріже останню цифру числа і запише її зліва.

1.108. Користувач вводить трицифрове число. Написати програму, яка вирізає з числа середню цифру і виводить двозначне число.

1.109. Користувач вводить чотирицифрове число. Написати програму, яка поміняє місцями першу і третю цифри, другу і четверту цифри.

1.110. Користувач вводить двоцифрове число. Програма повинна вивести різницю і добуток цифр числа.

1.111. Написати програму, яка знаходить різницю між першою і останньою цифрами трицифрового числа.

1.112. Користувач вводить чотирицифрове число. Програма повинна вивести різницю між першою і останньою цифрами.

1.113. Написати програму, яка виводить добуток першої і останньою цифр п'ятицифрового числа.

1.114. Користувач вводить п'ятицифрове число. Програма повинна визначити, у скільки разів остання цифра більша за другу.

1.115. Користувач вводить три двоцифрових числа. Програма повинна вивести добуток перших цифр цих чисел.

1.116. Користувач вводить два трицифрових числа. Програма повинна обчислити добуток їх середніх цифр.

1.117. Вантажний автомобіль має вантажопідйомність w т. Автомобіль завантажено мішками з просом, кожен масою 40 кг. Написати програму для обчислення кількості мішків, які може перевезти автомобіль.

1.118. Користувач вводить чотирицифрове число. Програма повинна вивести окремо суму двох перших і двох останніх цифр числа.

1.119. Користувач вводить чотирицифрове число. Програма повинна вивести окремо добуток двох перших і двох останніх цифр числа.

1.120. Написати програму, яка обчислить добуток двох середніх цифр чотирицифрового числа.

1.121. Написати програму, яка обчислить суму першої і останньої цифр п'ятицифрового числа.

1.122. Написати програму, яка обчислить добуток першої і останньої цифр чотирицифрового числа.

1.123. Написати програму, яка обчислить суму першої і другої цифр чотирицифрового числа.

1.124. Написати програму, яка визначить, скільки цілих фунтів поміщається в коробці місткістю F кг.

1.125. Написати програму, яка обчислить суму двох останніх цифр чотирицифрового числа.

1.126. Користувач вводить два двоцифрових числа. Програма повинна склеїти ці два число в одне чотирицифрове число. Оператори роботи з рядками не застосовувати.

1.127. Користувач вводить два чотирицифрових числа. Програма повинна обчислити суму двох чисел, кожне з яких отримано шляхом відкидання двох перших цифр чотирицифрового числа.

1.128. Користувач вводить два чотирицифрових числа. Програма повинна обчислити суму двох чисел, кожне з яких отримано шляхом відкидання двох останніх цифр чотирицифрового числа.

1.129. Написати програму, яка обчислює суму двох останніх цифр кожного з двох чотирицифрових чисел, введених з клавіатури.

1.130. Написати програму, яка обчислює добуток двох середніх цифр шестицифрового числа, введеного з клавіатури.

1.131. Користувач вводить чотирицифрове число. Програма повинна змінити порядок цифр у числі на протилежний. Оператори для роботи з рядками не застосовувати.

1.132. Написати програму, яка обчислює суму і різницю першої та останньої цифр чотирицифрового числа, введеного користувачем.

1.133. Написати програму, яка дописує до трицифрового числа Z в кінці цифру, введenu користувачем. Число Z також вводиться користувачем.

1.134. Користувач вводить чотирицифрове число. Програма повинна вписати середню цифру 8 і вивести отримане число на екран.

1.135. Користувач вводить трицифрове число. Програма повинна вилучити з нього середню цифру, а отримане число вивести на екран.

1.136. Користувач вводить чотирицифрове число. Програма повинна дописати в число на передостанню позицію цифру 7 і отримане число вивести на екран.

1.137. Користувач вводить шестицифрове число. Програма повинна вилучити з нього дві середні цифри і вивести їх на екран.

1.138. Користувач вводить п'ятицифрове число. Програма повинна вилучити середню цифру, а отримане чотирицифрове число вивести на екран.

1.139. Користувач вводить два трицифрових числа. Програма повинна поміняти місцями їх перші цифри і вивести отримані числа на екран.

1.140. Користувач вводить двоцифрове число. Програма повинна дописати в кінці цього числа останню цифру двічі.

1.141. Користувач вводить два двоцифрових числа. Програма повинна склеїти їх в одне число. Функції і методи роботи з рядками не використовувати.

1.142. Користувач вводить два трицифрових числа. Програма повинна склеїти ці числа в одне число і вивести на екран. Функції і методи роботи з рядками не використовувати.

1.143. Користувач вводить два трицифрових числа. Програма повинна обчислити добуток їх середніх цифр.

1.144. Користувач вводить двоцифрове число. Програма повинна обчислити добуток його цифр і дописати в кінці числа цифру 1.

1.145. Користувач вводить двоцифрове число. Програма повинна дописати в кінці числа цифру 5, а на початку числа – цифру 2.

1.146. Користувач вводить шестизначне число. Програма повинна розрізати це число на два тризначних. Функції і методи роботи з рядками не застосовувати.

1.147. Користувач вводить п'ятизначне число Z. Програма повинна вивести число, яке складається з двох останніх цифр числа Z.

1.148. Користувач вводить п'ятизначне число Z. Програма повинна вивести число, яке складається з двох перших цифр числа Z.

1.149. Написати програму, яка виводить на екран середні три цифри п'ятизначного числа, введеного з клавіатури.

1.150. Написати програму, яка обчислює добуток першої і останньої цифр чотирицифрового числа, введеного користувачем.

1.151. Користувач вводить трицифрове число. Програма повинна дописати в кінці цього числа таке саме число і результат вивести на екран. Функції і методи роботи з рядками не застосовувати.

1.152. Користувач вводить трицифрове число. Програма повинна записати його у зворотному порядку і дописати в кінці числа вихідне число, а результат вивести на екран. Функції і методи роботи з рядками не застосовувати.

1.153. З клавіатури вводиться двозначне число. Програма повинна в кінці цього числа дописати двічі це саме число, після чого вивести результат на екран. Функції і методи роботи з рядками не застосовувати.

1.154. Один з учнів грав на ігрових автоматах і виграв суму грошей X грн ($X < 100$). Виграш був виданий монетами вартістю 5, 1 і 2 грн. Написати програму, яка обчислить найменшу кількість монет, виданих у якості виграшу.

1.155. Фірма продала автомобілі марок «Фольксваген», «Ланос» і «Сенс» у співвідношення 5:3:2. Як відомо, кількість автомобілів виражається цілими числами. Написати програму, яка повинна обчислити: 1) найменшу загальну кількість автомобілів, яку продала фірма; 2) кількість автомобілів кожної марки, продану фірмою, якщо загальна кількість проданих автомобілів дорівнює Y .

1.156. Фірма WSW продає велосипеди і мопеди. У травні співвідношення проданих велосипедів і мопедів становило 2:3. Всього фірма продала X транспортних засобів. Написати програму, яка визначить кількість окремо велосипедів і окремо мопедів, проданих фірмою в травні. Число X вводиться користувачем.

1.157. Фірма ZZZ продає фарбу двох видів. У липні фірма продала x_1 банок фарби першого виду і x_2 банок фарби другого виду. Співвідношення $x_1:x_2=4/5$. Написати програму, яка обчислить загальну кількість банок фарби, проданих у липні, якщо відомо число x_1 (вводиться з клавіатури).

1.158. Фірма SKS випускає печиво двох марок, розфасоване у ящики. Кількість ящиків печива, випущених у червні, відноситься як 2/3. Написати програму, яка обчислить кількість ящиків кожної марки печива, якщо відомо загальну кількість ящиків печива Z (вводиться з клавіатури).

1.159. Написати програму, яка замінює значення двох змінних місцями.

1.160. Користувач вводить два трицифрових числа. Написати програму, яка поміняє місцями їх останні цифри.

1.161. Користувач вводить два трицифрових числа. Написати програму, яка поміняє місцями їх перші цифри.

1.162. Користувач вводить два трицифрових числа. Написати програму, яка поміняє місцями їх середні цифри.

1.163. Користувач вводить два двоцифрових числа: a_1 і a_2 . Програма повинна утворити з них чотирицифрове число, перша і

третьою цифрою якого є цифрами числа a_1 , а друга і четверта – цифрами числа a_2 .

1.164. Користувач вводить два трицифрових числа: a_1 і a_2 . Програма повинна утворити з них шестицифрове число, перша, третя і п'ята цифри якого є цифрами числа a_1 , а друга, четверта і шоста – цифрами числа a_2 .

1.165. Користувач вводить дві цифри. Програма повинна утворити з них два різних двоцифрових числа.

1.166. Користувач вводить два двоцифрових числа. Написати програму, яка утворює з них два різних чотирицифрових числа.

1.167. Написати програму, яка утворює двоцифрове число з двох випадкових цифр, відмінних від 0.

1.168. Написати програму, яка утворює трицифрове число з трьох випадкових цифр, відмінних від 0.

1.169. Написати програму, яка утворює трицифрове число з трьох випадкових цифр, перша з яких не може дорівнювати 0.

1.170. Користувач вводить два двоцифрових числа a_1 і a_2 . Програма повинна утворити з них двоцифрове число, взявши перші цифри з чисел a_1 і a_2 .

1.171. З клавіатури вводяться два двоцифрових числа x і y . Програма повинна утворити з їх останніх цифр двоцифрове число і вивести його на екран.

1.172. Користувач вводить два двоцифрових числа x і y . Написати програму, яка утворює з першої цифри числа x і другої цифри числа y двоцифрове число z і виводить його на екран.

1.173. З клавіатури вводиться два трицифрове число. Програма повинна обчислити добуток їх середніх цифр.

1.174. Користувач вводить два трицифрових числа. Програма повинна вивести на екран окремо добуток їх перших цифр і добуток їх останніх цифр.

1.175. Користувач вводить трицифрове число, цифри якого позначимо як a , b , c . Програма повинна вивести випадкове число в діапазоні $[a;b]$.

1.176. Користувач задає двоцифрове число x , програма генерує випадкове двоцифрове число y . Далі програма повинна обчислити число $z=x*y$.

1.177. З клавіатури вводяться два двоцифрових числа x і y . Програма повинна утворити з них двоцифрові числа a і b за таким принципом: 1) число a утворюється з перших цифр чисел x і y ; 2) число b утворюється з останніх цифр чисел x і y .

1.178. Користувач вводить чотирицифрове число. Програма повинна обчислити суму кубів усіх його цифр.

1.179. Написати програму, яка генерує два випадкових цілих числа у діапазоні $[1;x]$ і обчислює їх різницю. Число x задається з клавіатури.

1.180. Написати програму, яка обчислює добуток двох випадкових цілих чисел, перше з яких у діапазоні $[1;99]$, друге – у діапазоні $[100, 400]$.

1.181. Користувач задає чотирицифрове число. Програма повинна обчислити добуток першої і останньої цифр цього числа.

1.182. Написати програму, яка утворює чотирицифрове число шляхом склеювання двох випадкових двоцифрових чисел.

1.183. Користувач вводить чотирицифрове число A . Написати програму, яка замінить дві середні цифри числа A цифрами випадкового цілого числа у діапазоні $[10;99]$.

1.184. Користувач вводить трицифрове число M . Програма повинна замінити дві останні цифри числа M цифрами випадкового числа у діапазоні $[10;99]$.

1.185. Користувач задає трицифрове число K . Програма повинна замінити перші дві цифри числа K цифрами випадкового числа у діапазоні від 10 до 50.

1.186. Користувач задає трицифрове число x . Програма повинна замінити середню цифру числа випадковою.

1.187. Користувач задає п'ятицифрове число Z . Програма повинна згенерувати випадкове двоцифрове число і замінити ним перші дві цифри числа Z .

1.188. З клавіатури вводиться трицифрове число. Програма повинна замінити всі його цифри на випадкові цифри у діапазоні від 1 до 6.

1.189. Користувач вводить два двоцифрових числа x_1 і x_2 . Програма повинна утворити з них два чотирицифрові числа y_1 і y_2 такі, що y_1 складається з послідовно записаних чисел x_1 і x_2 , а число y_2 – з послідовно записаних чисел x_2 і x_1 . Функції для роботи з рядками і символами не використовувати.

1.190. Користувач вводить два двоцифрові числа x_1 і x_2 . Програма повинна утворити з них чотирицифрове число y , а потім згенерувати випадкове число z у діапазоні $[0;y]$. Функції для роботи з рядками і символами не використовувати.

1.191. Користувач вводить чотирицифрове число x . Програма повинна розбити його на два двоцифрових числа y_1 і y_2 , після чого згенерувати два випадкових числа z_1 і z_2 – перше в діапазоні $[0;y_1]$, друге – у діапазоні $[0;y_2]$, після чого вивести обидва числа на екран.

1.192. Користувач вводить трицифрове число x . Програма повинна розбити його на двоцифрове і одноцифрове, а потім знайти остачу від ділення двоцифрового числа на одноцифрове.

1.193. Написати програму, яка обчислює добуток цифр п'ятицифрового числа, введеного з клавіатури.

1.194. З клавіатури задається чисельник дробу. Програма повинна обчислити найменший знаменник дробу, такий, щоб дріб був правильним.

1.195. З клавіатури задається знаменник дробу. Програма повинна обчислити найбільший чисельник дробу, такий, щоб дріб був правильним.

1.196. Написати програму, яка обчислює суму двох випадкових цілих чисел у діапазоні від 0 до x . Число x вводиться з клавіатури.

1.197. Написати програму, яка обчислює суму квадратів двох випадкових чисел у діапазоні $[0;x]$. Число x вводиться з клавіатури.

1.198. Користувач вводить шестицифрове число. Програма повинна розбити це число на два трицифрових і вивести на екран їх різницю.

1.199. Написати програму, яка генерує два випадкових двоцифрових числа і утворює з них одне чотирицифрове.

1.200. Написати програму, яка обчислює різницю двох випадкових трицифрових чисел.

Задачі підвищеної складності

1.201. Гонорар адвоката чомусь виявився некруглою сумою x грн. У касі йому видали гонорар різними купюрами, які мають вартість 500, 200, 100, 50, 20, 20 і 5 грн, а також монетами вартістю 1 і 2 грн. Програма повинна обчислити кількість купюр, які отримав адвокат.

1.202. Існують монети по 1, 2, 5, 25, 50 коп. Написати програму, яка обчислить кількість монет для того, щоб віддати задачу по купцю. Задача виражена в копійках (від 1 до 99).

1.203. Одна з систем шифрування працює таким чином: спочатку генеруються два випадкових цілих числа a і b в діапазоні $[100;999]$, потім генерується третє випадкове ціле число в діапазоні $[a;b]$. Написати програму, яка реалізує даний алгоритм.

1.204. Один з алгоритмів шифрування реалізується у такій послідовності: 1) початкове чотиризначне число розбивається на два двоцифрових; 2) після кожного з двоцифрових чисел дописується випадкове двозначне число; 3) усі числа склеюються в одне восьмизначне число. Написати програму, яка реалізує даний алгоритм.

1.205. Алгоритм дешифрування має таку послідовність: 1) вхідне восьмизначне число розбивається на два чотиризначних; 2) від кожного з отриманих чотиризначних чисел відрізаються останні дві цифри; 3) два двоцифрові числа, що залишилися,

склеюються в одне чотирицифрове число. Написати програму, яка реалізує даний алгоритм.

1.206. Користувач вводить два двоцифрових натуральних числа. Написати програму, яка обчислить їх найменше спільне кратне обох чисел.

1.207. На площині міститься прямокутник зі сторонами $[a;b]$, де a і b – парні числа, задані користувачем. Центр перетину діагоналей співпадає з центром координат, а сторони прямокутника паралельні осям координат. Програма повинна згенерувати випадкові числа x і y – координати точки, яка знаходиться всередині прямокутника.

1.208. На площині зображено коло радіусом R (задається користувачем) з центром у початку координат. Програма повинна згенерувати координати точки $A(x; y)$, яка знаходиться всередині кола.

1.209. Користувач задає розміри прямокутника. Програма повинна обчислити площу кола, описаного навколо прямокутника.

1.210. Користувач задає площу квадрата. Програма повинна обчислити площу кола, описаного навколо квадрата.

1.211. Користувач задає площу кола. Програма повинна обчислити площу квадрата, вписаного у коло.

1.212. Користувач задає периметр кола. Програма повинна обчислити периметр квадрата, вписаного у коло.

1.213. З клавіатури задають периметр квадрата. Програма повинна обчислити периметр кола, описаного навколо квадрата.

1.214. Користувач вводить довжини основ і бічних сторін рівнобічної трапеції. Програма повинна обчислити площу трапеції.

1.215. З клавіатури задають довжину діагоналі прямокутника і кут між діагоналями в градусах. Програма повинна обчислити площу прямокутника.

1.216. Правильний шестикутник вписано у коло радіусом R . Програма повинна обчислити площу шестикутника.

1.217. Прямокутник із сторонами a і b вписано в коло. Написати програму, яка обчислить площу круга.

1.218. Велосипедне колесо діаметром D має N спиць. Написати програму, яка визначить відстань між кріпленням спиць на ободі колеса.

1.219. Маховик двигуна робить N обертів за хвилину. Написати програму, яка обчислить частоту обертання маховика.

1.220. З листового алюмінію виготовлено бак у вигляді прямокутного паралелепіпеда розмірами a, b, c . 1 м^2 листового алюмінію важить $0,9 \text{ кг}$. Написати програму, яка обчислить масу порожнього баку і баку з водою.

1.221. Бак, який має форму циліндра, виготовлено з листової сталі. Розміри бака: діаметр D , висота h . 1 м² листової сталі має масу 1,2 кг. Написати програму, яка обчислить масу порожнього бака і бака з водою.

1.222. Довжини сторін трикутника дорівнюють відповідно a, b, c . Написати програму, яка обчислить радіус кола, що має таку саму площу, як і трикутник.

1.223. Довжини сторін прямокутника дорівнюють a і b . Написати програму, яка обчислить радіус кола, яке має таку саму площу, що й прямокутник.

1.224. Правильний шестикутник має площу S . Навколо шестикутника описано коло. Написати програму, яка визначить, у скільки разів площа описаного кола більша за площу шестикутника.

1.225. У квадрат зі стороною a вписано коло. Програма повинна визначити, у скільки разів площа квадрата більша за площу круга.

Розділ 2. Алгоритмічна структура «Розгалуження»

У мові Python алгоритмічна конструкція «розгалуження» реалізується у вигляді:

```
if <логічна умова> :  
<блок операторів, які виконуються у тому випадку, коли умова  
справджується>  
else:  
<блок операторів, які виконуються у тому випадку, коли умова не  
справджується>
```

Складне розгалуження має такий вигляд:

```
if <логічна умова> :  
    <блок операторів, які виконуються у тому випадку, коли  
перша умова справджується>  
elif <друга логічна умова>:  
    <блок операторів, які виконуються у тому випадку, коли  
друга умова справджується>  
else:  
    <блок операторів, які виконуються у тому випадку, коли  
жодна умова не справджується>
```

Приклади задач

Приклад 2.1. Користувач вводить трицифрове число, у якому всі цифри різні. Написати програму, яка визначає – перша чи остання цифра числа більша.

Текст програми(if02)

```
x=int(input("Введіть число: "))  
last=x%10  
first=int(x/100)  
if first>last:  
    print("Перша цифра більша")  
else:  
    print("Остання цифра більша")
```

Приклад 2.2. Написати програму, яка порівнює два числа, введених з клавіатури.

Текст програми (if01)

```
a=float(input("Введіть число a: "))  
b=float(input("Введіть число b: "))  
if a>b:  
    print("Число a більше")  
elif b>a:  
    print("Число b більше")
```

```
else:  
    print("Числа a і b рівні")
```

Складна умова

У таких програмах оператор умови записується у вигляді

```
if x>3 and x<10:  
    <operator>
```

Умова АБО:

```
if x<4 or x>5:  
    <operator>
```

Приклад 2.3.

Написати програму, яка перевіряє,

Задачі для виконання

Проста умова

2.1. Користувач вводить з клавіатури два числа. Написати програму, яка визначає, яке число більше.

2.2. Написати програму, яка порівнює густини речовин, з яких виготовлено два бруска. Маса брусків і їх об'єми вводить користувач.

2.3. Написати програму, що визначатиме, яка з фігур має більшу площу – квадрат зі стороною a чи круг з радіусом R . Користувач вводить величини a і R .

2.4. Користувач вводить натуральне число. Написати програму, яка визначить, чи є це число парним.

2.5. Користувач вводить трицифрове число. Написати програму, яка порівняє першу та останню цифри числа.

2.6. Користувач вводить з клавіатури два натуральних числа – N і x . Написати програму, яка визначить, чи є число x дільником числа N .

2.7. Територія шкільного подвір'я, яка має форму прямокутника довжиною a м і шириною b м, огорожена дерев'яним парканом висотою 1,2 м. Для пофарбування паркану було закуплено x банок фарби по 2,8 кг у кожній. На фарбування 1 м^2 витрачається 200 г фарби. Програма повинна визначити, чи вистачить фарби для пофарбування паркану.

2.8. Написати програму, яка порівняє два випадкових числа (обидва – з проміжку $[0; 225]$).

2.9. Написати програму, яка перевіряє можливість існування трикутника з заданими кутами.

2.10. Написати програму, яка перевіряє можливість існування трикутника з заданими сторонами.

2.11. Користувач вводить з клавіатури два трицифрових числа. Програма повинна порівняти останні цифри цих чисел.

2.12. Користувач вводить з клавіатури чотирицифрове число. Програма повинна визначити, яка цифра – перша чи остання – більша.

2.13. Користувач вводить два трицифрових числа. Програма повинна порівняти середні цифри обох чисел.

2.14. Написати програму для порівняння площ квадрата і круга. Розміри фігур вводяться з клавіатури.

2.15. Користувач вводить трицифрове число. Програма повинна перевірити, чи є у його запису цифра 1.

2.16. Користувач вводить чотирицифрове число. Програма повинна перевірити, чи є у його запису цифра 7.

2.17. Користувач вводить два дійсних числа. Написати програму для порівняння їх квадратів.

2.18. Користувач вводить три цілих числа. Програма повинна обчислити, у скільки разів найбільше з них перевищує найменше.

2.19. Користувач вводить три дійсних числа. Програма повинна обчислити корені чисел, більших від 0.

2.20. Користувач вводить сторони прямокутного трикутника у довільній послідовності. Програма повинна перевірити можливість існування трикутника з цими сторонами.

2.21. Користувач вводить три двоцифрових числа. Програма повинна порівняти останні цифри цих чисел.

2.22. Користувач вводить два чотирицифрових числа. Програма повинна вивести число, сума цифр якого більша.

2.23. Користувач вводить два трицифрових числа. Програма повинна вивести число, добуток цифр якого найменший.

2.24. Користувач вводить два п'ятицифрових числа. Програма повинна порівняти добутки перших двох цифр для обох чисел.

2.25. Користувач вводить три трицифрових числа. Програма повинна вивести число, середня цифра якого найбільша.

2.26. Користувач вводить 4 натуральних числа. Програма повинна вивести окремо найбільше і найменше числа.

2.27. Користувач вводить натуральне число. Програма повинна перевірити, чи ділиться це число на 17.

2.28. Написати програму для порівняння об'ємів двох куль. Діаметри куль задаються з клавіатури.

2.29. Написати програму для порівняння об'ємів двох кубів. Розміри кубів задаються з клавіатури.

2.30. Користувач вводить з клавіатури трицифрове число. Програма повинна визначити, яка цифра числа більша.

2.31. Користувач вводить чотирицифрове число. Програма повинна визначити, яка з двох цифр – перша чи остання – менша.

2.32. Користувач вводить чотирицифрове число. Програма повинна визначити, сума яких чисел більша – двох перших чи двох останніх.

2.33. Користувач вводить чотирицифрове число. Програма повинна отримати з нього число, яке складається з двох середніх цифр, визначити, чи ділиться отримане число на 3.

2.34. Написати програму, яка визначить, чи є парною перша цифра трицифрового числа, введеного користувачем.

2.35. Користувач вводить чотирицифрове число. Програма повинна визначити, чи ділиться на 3 остання цифра числа.

2.36. Написати програму, яка визначає подільність першої цифри чотирицифрового числа на 2.

2.37. Користувач вводить чотирицифрове число. Програма повинна визначити, чи ділиться на 4 сума цифр числа.

2.38. З клавіатури вводиться трицифрове число. Програма повинна визначити, чи ділиться середня цифра цього числа на 2.

2.39. Користувач вводить два двоцифрових числа. Програма повинна вивести число, сума цифр якого найбільша.

2.40. Користувач вводить два двоцифрових числа. Програма повинна вивести число, сума цифр якого найменша.

2.41. Користувач вводить два двоцифрових числа. Програма повинна вивести число, добуток цифр якого найбільший.

2.42. Користувач вводить два двоцифрових числа. Програма повинна вивести число, добуток цифр якого найменший.

2.43. З клавіатури вводяться два трицифрових числа. Програма повинна вивести число, добуток цифр якого більший.

2.44. З клавіатури вводяться два трицифрових числа. Програма повинна вивести число, добуток цифр якого менший.

2.45. Користувач вводить чотирицифрове число. Написати програму, яка виведе число, друга цифра якого більша.

2.46. Користувач вводить з клавіатури розміри двох прямокутних паралелепіпедів. Програма повинна вивести на екран розміри паралелепіпеда, об'єм якого більший.

2.47. Користувач вводить з клавіатури сторони двох трикутників. Програма повинна вивести на екран розміри трикутника, площа якого більша.

2.48. Користувач вводить діагоналі двох ромбів. Програма повинна вивести периметр ромба, площа якого найбільша.

2.49. Написати програму для порівняння площ бічних поверхонь циліндра і прямокутного паралелепіпеда. Розміри циліндра і паралелепіпеда вводяться з клавіатури.

2.50. Користувач вводить координати двох векторів на площині. Програма повинна вивести координати того вектора, який має більшу довжину.

Складна умова

2.51. Користувач задає координати точки x на площині. Написати програму, яка виводить номер чверті декартової площини, в якій знаходиться точка. Всі координати точки відмінні від 0.

2.52. Користувач вводить коефіцієнти a , b , c квадратного рівняння. Написати програму, яка виводить корені рівняння або повідомлення, що коренів не існує.

2.53. Написати програму, яка визначає, чи ділиться введене користувачем ціле число одночасно на 7 і 9.

2.54. Користувач вводить число. Програма повинна визначити, чи ділиться це число одночасно на 3 і 5.

2.55. Користувач вводить з клавіатури чотири числа. Програма повинна визначити найбільше з цих чисел.

2.56. Написати програму, яка визначить найменше з чотирьох чисел.

2.57. Користувач вводить трицифрове число. Програма повинна вивести найменшу цифру числа.

2.58. Користувач вводить трицифрове число. Програма повинна вивести найбільшу цифру числа.

2.59. Написати програму, яка визначить середнє число з трьох.

2.60. Написати програму, яка визначить найбільше з трьох чисел.

2.61. Користувач вводить сторони трикутника. Програма повинна визначити, чи може існувати трикутник з такими сторонами.

2.62. Написати програму, яка повинна визначити можливість існування чотирикутника з заданими кутами.

2.63. Користувач вводить чотири цілих числа. Програма повинна вивести найбільше і найменше з них.

2.64. Користувач вводить три дійсних числа. Програма повинна вивести число, модуль якого найбільший.

2.65. Користувач вводить координати точки на площині. Програма повинна вказати, вище чи нижче від осі x знаходиться точка, справа чи зліва від осі y знаходиться точка.

2.66. Написати програму, яка перевіряє, чи належить введене користувачем число X проміжку $[a;b]$. Всі три числа вводяться з клавіатури.

2.67. Написати програму, яка перевіряє, чи виходить введене користувачем число z за межі діапазону $[x_1;x_2]$. Число z вводиться користувачем, змінні x_1 і x_2 ініціалізуються на початку програми.

2.68. Користувач вводить два трицифрових числа. Програма повинна перевірити, чи діляться обидва числа на x (задається на початку програми).

2.69. На координатній площині розміщено прямокутник так, що його сторони паралельні осям координат, а центр перетину діагоналей співпадає з центром координат. Розміри прямокутника задано на початку програми (5×3). Користувач задає координати $(x; y)$ точки на площині. Програма повинна визначити, чи належить точка прямокутнику.

2.70. Користувач задає координати чотирьох вершин чотирикутника. Програма повинна визначити, чи є цей чотирикутник ромбом.

2.71. Користувач задає координати чотирьох вершин чотирикутника. Програма повинна визначити, чи є цей чотирикутник паралелограмом.

2.72. Користувач задає довжини сторін трикутника. Програма повинна визначити: 1) можливість існування трикутника з цими довжинами сторін; 2) чи є трикутник рівнобедреним; 2) чи є трикутник рівностороннім.

2.73. Користувач задає координати вершин трикутника на площині. Програма повинна визначити, чи є даний трикутник рівнобедреним.

2.74. Користувач задає довжини сторін трапеції у довільному порядку. Програма повинна визначити, чи може бути така трапеція рівнобедреною.

2.75. Користувач вводить довжини трьох ребер прямокутного паралелепіпеда. Програма повинна визначити, чи є цей паралелепіпед кубом.

2.76. Користувач вводить довжини трьох ребер прямокутного паралелепіпеда у довільному порядку. Програма повинна визначити, чи є основа паралелепіпеда квадратом.

2.77. Написати програму, яка визначає найбільшу цифру у запису чотирицифрового числа, введеного з клавіатури.

2.78. Користувач вводить координати вершин трикутника на площині. Програма повинна вивести найменшу сторону трикутника.

2.79. На площині проведено дві прямі через центр координат під кутом 45° і 135° до осі x . Користувач задає координати точки $B(x; y)$. Програма повинна визначити, чи належить точка хоча б одній з прямих.

2.80. Густина води 1 г/см^3 . Густина бензину – $0,78 \text{ г/см}^3$. Користувач вводить густину твердого тіла, зануреного по черзі у кожну з рідин. Програма повинна визначити, чи буде плавати тіло в цих рідинах.

2.81. Користувач вводить три трицифрових числа. Програма повинна вивести число, остання цифра якого найбільша.

2.82. Користувач вводить окремо сторону квадрата і дві сторони прямокутника. Програма повинна визначити фігуру, площа якої найбільша.

2.83. Користувач вводить окремо три пари чисел – довжини основ трьох трапецій. Програма повинна визначити трапецію, середня лінія якої найбільша.

2.84. Коло радіусом R_1 має координати центра $(0;0)$, а коло з радіусом R_2 – координати центра $(7;9)$. Радіуси кіл вводить користувач. Програма повинна визначити кількість точок перетину кіл.

2.85. Коло радіусом R_1 має центр на осі x у точці $(-6;0)$, а коло радіусом R_2 – на осі x у точці $(12;0)$. Написати програму, яка визначить кількість точок перетину двох кіл. Радіуси кіл вводяться з клавіатури.

2.86. Коло радіусом R_1 має центр на осі x у точці $(2;0)$, а коло радіусом R_2 – на осі x у точці $(0;16)$. Написати програму, яка визначить кількість точок перетину двох кіл. Радіуси кіл вводяться з клавіатури.

2.87. Користувач вводить число. Програма повинна порівняти його з двома випадковими числами в діапазоні $[1; 200]$.

2.88. Користувач вводить три трицифрових числа. Програма повинна обчислити суму двох більших чисел.

2.89. Користувач вводить три трицифрових числа. Програма повинна обчислити суму двох менших чисел.

2.90. Прямокутник довжиною a і висотою b розміщений на координатній площині так, що центр перетину діагоналей співпадає з центром координат, а сторони паралельні осям координат. З точки $M(x; y)$ проведено горизонтальну пряму. Програма повинна визначити, чи перетинає пряма сторони прямокутника. Координати точки M вводить користувач.

2.91. Прямокутник довжиною a і висотою b розміщений на координатній площині так, що центр перетину діагоналей співпадає з центром координат, а сторони паралельні осям координат. З точки $M(x; y)$ проведено вертикальну пряму. Програма повинна визначити, чи перетинає пряма сторони прямокутника. Координати точки M вводить користувач.

2.92. Два кола радіусом R кожне розміщені на координатній площині так, що їх центри знаходяться на осі x у точках $(x_1; 0)$ і $(x_2; 0)$. Написати програму, яка визначає кількість точок перетину цих кіл.

2.93. Користувач вводить чотири числа. Програма повинна визначити, чи є серед цих чисел два однакових. Алгоритмічну конструкцію «цикл» не використовувати.

2.94. Користувач вводить чотири числа. Програма повинна вивести на екран два більших числа. Алгоритмічну конструкцію «цикл» не використовувати.

2.95. Користувач вводить три числа. Програма повинна вивести на екран два менших з них.

2.96. Користувач вводить чотири числа. Програма повинна вивести добуток двох менших із них.

2.97. Користувач вводить три числа. Програма повинна вивести на екран суму двох менших з них.

2.98. Користувач вводить два числа. Програма повинна визначити, чи є вони обидва додатніми, чи є вони обидва від'ємними.

2.99. Користувач вводить три числа a , b , c . Програма повинна визначити, чи розташовані вони в порядку зростання ($a < b < c$).

2.100. Написати програму, яка повинна визначити знак добутку трьох чисел (a , b , c).

Задачі підвищеної складності

2.101. Прямокутник розміщено так, що його центр співпадає з центром координат, а його сторони паралельні осям координат. Користувач задає координати однієї з сторін прямокутника. Написати програму, яка виводить у три рядки координати x та y трьох інших вершин прямокутника.

2.102. Ромб розміщений на площині так, що його діагоналі лежать на осях координат, а точка їх перетину співпадає з центром координат. Користувач задає координати двох сусідніх вершин ромба. Програма повинна вивести координати двох інших вершин ромба.

2.103. На координатній площині зображено кільце внутрішнім діаметром d і шириною t так, що центр кільця співпадає з центром координат. Програма повинна визначити, чи належить точка $A(x;y)$ кільцю. Розміри кільця і координати точки вводяться з клавіатури.

2.104. На координатній площині зображено квадрат, діагоналі якого лежать на осях координат, а точка перетину діагоналей співпадає з центром координат. Довжина сторони квадрата задається на початку програми. Користувач вводить координати точки $F(x;y)$. Програма повинна визначити, чи належить точка квадрату.

2.105. Користувач задає координати центра і радіус першого кола, координати центра і радіус другого кола. Програма повинна визначити кількість точок перетину цих кіл.

2.106. Центр кола радіусом R співпадає з центром координат. Написати програму, яка визначить, чи перетинає це коло пряма $y=3x+2$. Радіус кола вводиться з клавіатури.

2.107. Після того, як у будинку було встановлено двотарифний електролічильник, визначати плату стали так: денний тариф (з 6.00 до 22.00) – 1 грн/кВт-год, нічний тариф (з 20.00 до 6.00) – 60 коп за кВт-год. Написати програму, яка обчислить вартість електроенергії величиною F кВт-год, спожитої у годину X . Числа X , F вводять користувач.

2.108. Квадрат зі стороною a вписано у коло. Написати програму, яка визначить, чи може коло мати радіус R . Числа a , R вводяться користувачем.

2.109. Квадрат має сторону a , радіус кола становить R . Програма повинна визначити, чи може коло бути вписаним у квадрат.

2.110. Сторони прямокутника мають довжину a , b . Радіус кола становить R . Програма повинна визначити, чи може бути даний прямокутник вписаним у це коло. Величини a , b , R вводяться користувачем.

2.111. Користувач задає три сторони трикутника. Програма повинна перевірити можливість існування трикутника з такими сторонами і обчислити три висоти трикутника.

2.112. Написати програму, яка виведе на екран назву фігури, площа якої більша – прямокутника зі сторонами a і b чи правильного шестикутника зі стороною c . Числа a , b , c вводяться з клавіатури.

2.113. Існують дві фігури – правильний восьмикутник зі стороною a і правильний шестикутник зі стороною b . Написати програму, яка визначить, площа якої фігури більша.

2.114. Існують дві фігури – коло радіусом R і правильний восьмикутник зі стороною a . Програма повинна назву фігури, площа якої більша. Значення a , R вводять користувач.

2.115. Користувач задає дві пари значень – розміри двох рівнобедрених трикутників (основи a_1 і a_2 , бічні сторони b_1 і b_2). Програма повинна вивести площу більшого трикутника.

2.116. Існують два геометричних тіла: прямокутний паралелепіпед з довжинами ребер a, b, c і куб з довжиною ребра x . Програма повинна визначити фігуру, площа поверхні якої більша.

2.117. Користувач вводять чотири натуральних числа. Програма повинна вивести найбільше і найменше числа. Масиви не використовувати.

2.118. Користувач задає три натуральних числа. Програма повинна вивести різницю між найбільшим і найменшим числами.

2.119. Користувач задає номер дня тижня. Програма повинна вивести назву дня.

2.120. Дерев'яний брусок у формі прямокутного паралелепіпеда має розміри $axbxc$. Написати програму, яка визначить, чи можна затулити цією цеглиною вибоїну у стіні, яка має розміри fxk .

2.121. Два кола радіусами $R1$ і $R2$ розташовані на одній прямій так, що відстань між центрами кіл дорівнює L . Програма повинна визначити кількість точок перетину кіл. Якщо кола розміщені одне всередині іншого, то програма повинна вивести -1 .

2.122. Точку M на площині задано координатами x і y (ініціалізуються на початку програми). Користувач задає координати точок A і B на площині. Програма повинна вивести назву і координати точки, яка знаходиться ближче до точки M .

2.123. Точка F на координатній прямій задана координатою x . Користувач задає координати точок A , B і C , які лежать на цій самій прямій. Програма повинна вивести назву точки, яка лежить найдалі від точки F .

2.124. Пряму задано рівнянням $y=-2x-4$. Користувач задає координати точки A . Програма повинна вивести повідомлення про те, вище чи нижче від прямої буде знаходитися точка A .

2.125. Графік функції задано рівнянням $y=6x+4$. Користувач задає координати точки C . Програма повинна визначити, вище чи нижче прямої знаходиться точка C .

Розділ 3. Алгоритмічна структура «Цикл»

У Python може бути кілька типів циклів. Розглянемо їх детальніше.

Цикл з лічильником

Цикл з параметром (лічильником) має вигляд:

```
for i in range (n):
```

<тіло циклу>

Зміст циклу звучить так: повторювати від i до n разів. У цьому ж виразі i – початкове значення лічильника, n – кінцеве значення лічильника.

Кількість повторів може бути сталою величиною, а може бути змінною і визначатись користувачем.

Отже, цикл з лічильником застосовується у тому випадку, коли кількість повторів можна визначити наперед.

Проте цикл з лічильником у Python відрізняється від такого ж циклу в інших мовах програмування. Він завжди починається з 0 і повторюється стільки разів, яке значення має змінна n .

Проте можна змінити

Приклади задач

Приклад 3.1. Вивести на екран у стовпчик усі цілі числа від 1 до n .

```
n=int(input("Введіть ціле число"))
for i in range (n):
    print(i)
```

Приклад 3.2. Вивести на екран квадрати усіх цілих чисел від a до b

```
from math import pow
a=int(input("Введіть a: "))
b=int(input("Введіть b: "))
x=0
for i in range(b):
    if i>a:
        x=pow(i,2)
        print(x)
```

Завдання для виконання

3.1. Написати програму, яка виводить квадрати усіх цілих чисел на проміжку від a до 100.

3.2. Написати програму, яка виводить таблицю вартості одиниць товару від 1 до 10. Ціну одиниці товару задає користувач.

3.3. Написати програму для переведення відстані з футів у сантиметри (1 фут=30,3 см). Таблиця повинна мати межі від 3 до 20 футів.

3.4. Написати програму, яка виводить 10 випадкових чисел у діапазоні $[a;b]$. Величини a і b задаються користувачем.

3.5. Написати програму, яка виводить на екран усі парні числа на проміжку від 0 до x . Число x задає користувач.

3.6. Користувач вводить число x . Програма повинна вивести на екран усі дільники числа 885, які є на проміжку $[1;x]$.

3.7. Користувач вводить числа Z і Y . Програма повинна вивести всі числа на проміжку від 0 до Z , кратні Y .

3.8. Користувач вводить числа A і B (обидва – більші за 25). Програма повинна вивести всі дільники цих чисел на проміжку від 1 до 25.

3.9. Користувач вводить натуральне число t . Програма повинна вивести всі числа, кратні 7, на проміжку від 0 до t^2 .

3.10. Користувач вводить натуральне число N . Програма повинна вивести всі числа на проміжку $[0; N]$, які діляться одночасно на 2 і на 7.

3.11. Написати програму, яка виводить всі числа, кратні 3 і 2, на проміжку від x_1 до x_2 . Числа x_1 і x_2 вводяться з клавіатури.

3.12. Користувач вводить двозначне натуральне число N . Програма повинна обчислити добуток усіх парних чисел в діапазоні від 1 до N , кратних 7.

3.13. Користувач вводить двозначне натуральне число N . Програма повинна обчислити середнє арифметичне всіх парних чисел від N до 99.

3.14. Написати програму, яка обчислює середнє арифметичне всіх трицифрових натуральних чисел від 200 до N . Число N вводиться з клавіатури.

3.15. Написати програму, яка обчислює середнє геометричне 5 послідовних двоцифрових натуральних чисел, починаючи з R . Число R вводиться з клавіатури.

3.16. Користувач вводить число Q . Програма повинна визначити його найбільший і найменший дільники.

3.17. Написати програму, яка виводить квадратні корені всіх натуральних чисел, починаючи від 111 до числа L . Число L вводиться з клавіатури.

3.18. Написати програму, яка обчислює суму квадратів перших 10 парних натуральних чисел.

3.19. Написати програму, яка обчислює суму цифр всіх натуральних чисел на проміжку від a до b . Числа a і b вводяться з клавіатури.

3.20. Написати програму, яка виводить на екран всі натуральні числа до N , які одночасно діляться на 3 і закінчуються на цифру 1.

3.21. Написати програму, яка виводить на екран перші 6 парних чисел, які діляться на 7.

3.22. Написати програму, яка виводить таблицю значень мір довжини в ярдах і метрах від 0 до 20 ярдів.

3.23. Написати програму, яка виводить на екран таблицю переведення маси з фунтів в кілограми в межах від 0 до 20 фунтів.

3.24. Написати програму, яка виводить на екран таблицю для переведення мір довжини з футів в метри.

3.25. Написати програму, яка виводить на екран N рядків по N зірочок у кожному.

3.26. Користувач вводить число N – загальну кількість зірочок, яку програма повинна вивести на екран, записану в x рядків. Програма повинна визначити, на якому рядку тексту зупиниться виведення.

3.27. Написати програму, яка виводить таблицю квадратів і кубів усіх натуральних чисел на проміжку від x_1 до x_2 . Числа x_1 і x_2 задаються користувачем.

3.28. Написати програму, яка обчислює суму всіх чисел, які закінчуються цифрою 3, на проміжку від x_1 до x_2 . Числа x_1 і x_2 вводяться з клавіатури.

3.29. Користувач вводить число u . Програма повинна вивести у стовпчик усі числа, кратні u , на проміжку від 100 до 200.

3.30. Користувач задає число w ($w < 20$). Програма повинна вивести суму всіх двоцифрових чисел, кратних w .

3.31. Написати програму, яка виведе таблицю множення для чисел від 11 до 20.

3.32. Написати програму, яка виведе таблицю множення для чисел від 21 до 30.

3.33. Написати програму, яка виведе таблицю косинусів кутів від 0 до x градусів. Число x вводиться з клавіатури.

3.34. Написати програму, яка виведе таблицю тангенсів кутів від 1 до x градусів. Число x вводиться з клавіатури.

3.35. Користувач вводить парне число x . Програма повинна обчислити суму 8 послідовних парних чисел, більших за x .

3.36. Написати програму, яка виводить у першому рядку x символів 'a', у другому рядку – у символів 'b'. Числа x і y задаються з клавіатури.

3.37. Написати програму, яка виводить таблицю значень функції $y = x^3 - 4$ у межах від 0 до 15.

3.38. Користувач вводить межі діапазону $[x_1; x_2]$. Програма повинна вивести на екран квадрати і квадратні корені чисел, кратних 3.

3.39. Написати програму, яка виводить числа з проміжку $[x_1; x_2]$, кратні 3 і не кратні 2. Числа x_1 і x_2 вводяться користувачем.

3.40. Написати програму, яка обчислює середнє арифметичне всіх парних чисел, не кратних 5, на проміжку $[x_1; x_2]$. Числа x_1 і x_2 вводяться з клавіатури.

3.41. Написати програму, яка підраховує кількість цифр 2 у запису трицифрових натуральних чисел у діапазоні $[a; b]$. Числа a і b вводяться з клавіатури.

3.42. Написати програму, яка підраховує суму цифр всіх трицифрових натуральних чисел, кратних x , на проміжку від 122 до 197. Число x задається з клавіатури.

3.43. Написати програму, яка знаходить на проміжку $[a; b]$ трицифрове число, сума цифр якого найменша.

3.44. Написати програму, яка обчислює суму n випадкових цілих чисел у діапазоні $[0; 100]$. Число n задає користувач.

3.45. Написати програму, яка обчислює добуток трьох випадкових чисел у діапазоні $[0; x]$. Число x вводиться з клавіатури.

3.46. Написати програму, яка обчислює суму натуральних чисел від 1 до N . Число N – випадкове число з проміжку $[10; 35]$.

3.47. Написати програму, яка обчислює суму 7 випадкових чисел на проміжку від 10 до N , де N – випадкове число на проміжку $[15; 30]$.

3.48. Написати програму, яка виводить на екран послідовність з 11 чисел, кожне з яких дорівнює $i \cdot x$, де x – випадкове число в діапазоні $[10; 20]$.

3.49. Написати програму, яка виводить на екран N добутків $x_1 \cdot x_2$, де x_1 – випадкове число в діапазоні $[1; 10]$, x_2 – випадкове число в діапазоні $[31; 40]$. Число N задається з клавіатури.

3.50. Кожна сторінка книги налічує z рядків по w символів у кожному. Число z – випадкове число в межах від 30 до 35, число w – випадкове число в межах від 45 до 55. Програма повинна обчислювати загальну кількість символів на сторінці.

3.51. Написати програму, яка виводить таблицю значень функції $y = \frac{1}{\sqrt{x}}$ для всіх натуральних чисел від 11 до N . Число N задається з клавіатури.

3.52. Написати програму, яка виводить таблицю значень x і функції $x \cdot \text{suma_cifr}(x)$, де x – число, suma_cifr – сума цифр цього числа. Діапазон обчислень – $[x_1; x_2]$ – вводиться з клавіатури.

3.53. Користувач вводить з клавіатури 7 натуральних чисел. Написати програму, яка обчислює суму і середнє арифметичне введених чисел.

3.54. Користувач вводить 8 двоцифрових натуральних чисел. Програма повинна визначити, наскільки часто зустрічається цифра 7 у запису цих чисел.

3.55. Написати програму, яка виводить на екран 5 добутоків останніх цифр трицифрових випадкових чисел x_1 і x_2 . Число x_1 – у діапазоні [100;155], число x_2 – у діапазоні [200;280].

3.56. З клавіатури вводяться 6 чисел. Програма повинна визначити найменше і найбільше з цих чисел.

3.57. Користувач вводить значення опадів за кожний з днів тижня. Програма повинна обчислити середню кількість опадів за день.

3.58. Написати програму, яка виведе число x у стовпчик x разів. Число x вводиться з клавіатури.

3.59. Користувач вводить цифру. Програма повинна вивести на екран усі двозначні числа, які починаються на цю цифру.

3.60. Користувач вводить цифру. Програма повинна вивести на екран усі тризначні числа, які починаються на цю цифру.

3.61. Користувач вводить цифру. Програма повинна вивести на екран усі двозначні числа, які закінчуються на цю цифру.

3.62. Користувач вводить цифру. Програма повинна вивести на екран усі тризначні числа, які закінчуються на цю цифру.

3.63. Написати програму, яка виводить на екран усі двозначні числа, сума цифр яких ділиться на 5.

3.64. Написати програму, яка виводить на екран усі тризначні числа, сума цифр яких ділиться на 7.

3.65. Стіна має розміри $a \times b$ м. Визначити кількість цілих плиток розмірами 15×15 см, які потрібні для покриття стіни.

3.66. Користувач вводить 2 тризначні числа – a і b . Програма повинна обчислити суму всіх чисел, які містяться між числами a і b і друга цифра яких дорівнює 2, попередньо визначивши більше і менше з цих чисел.

3.67. Користувач вводить менше з двох чисел. Програма повинна обчислити більше число, якщо для його отримання необхідно менше число збільшити за допомогою x кроків, кожен з яких дорівнює v .

3.68. Бочка містить x літрів води. Хлопчик вичерпав з неї воду відром місткістю y літрів. Написати програму, яка виведе на екран об'єм води, яка залишилась у відрі після і вечерпувань.

3.69. В елеватор завантажували зерно. Першого дня завантажили 3 т, другого дня – 5 т, третього дня – 7 т тощо. Написати

програму, яка визначить, в який день кількість зерна в елеваторі перевищить z .

3.70. У кузов автомобіля було засипано x кг піску. Першого дня з нього відсипали 5 кг, другого дня – 8 кг, третього дня – 11 кг і т.д. Написати програму, яка визначить кількість днів, коли з кузова буде висипано весь пісок.

3.71. Написати програму, яка обчислює суму всіх двозначних чисел, кратних 8.

3.72. Написати програму, яка обчислює суму всіх тризначних чисел, кратних 29.

3.73. Користувач вводить числа a і b . Програма повинна визначити, скільки цифр містять числа у цьому діапазоні (одно-, дво- чи тризначні числа?).

3.74. Користувач вводить одноцифрове число X . Програма повинна вивести на екран квадрат з X цифр X .

3.75. Написати програму, яка виводить на екран X цифр X через пробіл.

3.76. Написати програму, яка обчислить кількість трицифрових чисел, що діляться на 19 без остачі.

3.77. Написати програму, яка виводить на екран у порядку спадання всі двоцифрові числа, кратні m . Число m вводиться користувачем.

3.78. Написати програму, яка виводить на екран у порядку спадання всі трицифрові числа, цифри яких однакові.

3.79. Написати програму, яка виводить на екран у порядку зростання всі двоцифрові числа, обидві цифри яких однакові.

3.80. Написати програму, яка виводить на екран усі двоцифрові числа, друга цифра яких більша за першу.

3.81. Написати програму, яка виведе на екран усі чисельники правильного дробу із знаменником Z . Число Z зчитати з клавіатури.

3.82. Користувач вводить натуральне число x . Програма повинна визначити кількість дільників цього числа.

3.83. Користувач вводить двозначне число x . Програма повинна визначити кількість двозначних дільників цього числа.

3.84. Написати програму, яка визначить кількість двоцифрових чисел – дільників числа x . Трицифрове число x задається з клавіатури.

3.85. Користувач вводить чотиризначне число Z . Програма повинна вивести на екран усі двоцифрові дільники числа Z .

3.86. Користувач вводить двоцифрове число a . Програма повинна вивести на екран усі трицифрові числа, кратні a .

3.87. Користувач вводить двоцифрове число a . Програма повинна вивести на екран усі чотирицифрові числа, кратні a .

3.88. Користувач вводить трицифрове число x . Програма повинна вивести у порядку спадання всі чотирицифрові числа, кратні x .

3.89. Користувач вводить чотиризначне число. Програма повинна вивести на екран усі трицифрові дільники цього числа.

3.90. Написати програму, яка виводить на екран усі двоцифрові дільники чотирицифрового числа, введеного користувачем.

3.91. Написати програму, яка виводить на екран усі трицифрові дільники чотирицифрового числа Z , введеного користувачем.

3.92. Розробити програму, яка виведе усі числа, кратні 4, які закінчуються цифрою 2, у діапазоні $[a;b]$, де a і b – випадкові цілі числа у діапазоні від 100 до 999.

3.93. Написати програму, яка генерує межі діапазону $[a;b]$, а потім – 9 випадкових цілих чисел у цьому діапазоні і знаходить їх суму.

3.94. Написати програму, яка визначає кількість дільників випадкового трицифрового числа Z . Дільники повинні бути відмінними від 1 і Z .

3.95. Написати програму, яка обчислює суму дільників числа Y , введеного користувачем. Дільники повинні бути відмінними від 1 і Y .

3.96. У числовій послідовності перше число 1, кожне наступне число на 3 більше за попереднє. Написати програму, яка виведе на екран N членів послідовності у стовпчик.

3.97. Перше число послідовності дорівнює 1,5, кожне наступне число більше за попереднє на 1,9. Написати програму, яка виведе на екран N чисел, які входять у послідовність.

3.98. Числова послідовність складається з чисел 2, 4,5, 7, 9,5... Написати програму, яка виведе у стовпчик перші N чисел послідовності, де N – випадкове число у діапазоні від 10 до 20.

3.99. Користувач задає два цілих числа – a_1 і a_2 . Програма повинна вивести на екран усі числа в діапазоні $[a_1;a_2]$, які діляться на 7 і закінчуються на 1.

3.100. Написати програму, яка виводить на екран таблицю множення на 24.

Цикл з передумовою

Цикл з передумовою має вигляд:

```
while <логічний вираз>:  
    <тіло циклу>
```

Наприклад:

```
while x>1:  
    x=x//10
```

Приклад

Вивести на екран натуральні числа, які не більші за n. Число n задає користувач.

```
n=int(input("Введіть n"))  
i=1;  
while i<n:  
    print(i)  
    i=i+1
```

Приклад

Користувач вводить будь-яке число в межах від 10 до 100000. Програма повинна вивести всі цифри числа у стовпчик.

```
n=int(input("Input n"))  
i=1.0  
x=1.0  
while n>0:  
    x=n%10  
    print(x)  
    n=n//10
```

Завдання для виконання

3.101. Вивести на екран всі непарні числа з інтервалу [a; b]. Оператор циклу з параметром не використовувати.

3.102. Вивести на екран найменше число, більше за 200, яке націло ділиться на k. Число k вводить користувач.

3.103. Написати програму, яка визначає максимальне число в інтервалі від a до b, яке є квадратом натурального числа.

3.104. Написати програму, яка визначає перше натуральне число, квадрат якого більший за задане число n.

3.105. Створити програму, яка знаходить суму непарних чисел, введених користувачем. Якщо користувач вводить парне число, програма виводить повідомлення про помилку. Якщо користувач вводить 0, програма припиняє роботу і виводить результат.

3.106. Написати програму, яка запитує у користувача 10 чисел і виводить їх суму.

3.107. Написати програму, яка виводить піраміду:

```
*  
**  
***  
****
```

Кількість рядків задає користувач.

3.108. Написати програму, яка виводить у стовпчик значення x та y для функції $y=x^2+x$. Значення x повинні змінюватися з кроком 0,2. Інтервал $[a; b]$ задає користувач.

3.109. Користувач вводить натуральне число, кількість цифр якого наперед невідома. Написати програму, яка знаходить суму цифр числа.

3.110. Написати програму, яка обчислює суму всіх цілих чисел у межах $[a;b]$, кратних 7. Числа a і b задає користувач. Цикл з параметром не використовувати.

3.111. Користувач вводить натуральне число. Написати програму, яка знаходить суму квадратів його цифр.

3.112. Користувач вводить натуральне число n . Вивести на екран найбільший дільник цього числа.

3.113. Користувач вводить два натуральних числа. Написати програму, яка перевірить, чи будуть ці два числа взаємно простими.

3.114. Написати програму, яка визначає кількість цифр 3 у натуральному числі, введеному користувачем.

3.115. Написати програму, яка виводить на екран всі числа з проміжку $[a;b]$, які є квадратами цілих чисел. Числа a і b задаються користувачем.

3.116. Написати програму, яка перевіряє введене користувачем число: чи є воно простим.

3.117. Користувач вводить натуральне число N . Написати програму, яка виводить на екран ті натуральні числа, квадрат яких не перевищує N . Цикл з лічильником не використовувати.

3.118. Користувач вводить натуральне число. Написати програму, яка виводить найбільшу цифру числа.

3.119. Користувач вводить натуральне число. Написати програму, яка виводить найменшу цифру числа.

3.120. Написати програму, яка виводить на екран всі натуральні числа, які діляться на 3, на інтервалі від a до b . Числа a і b вводяться користувачем. Цикл з параметром не використовувати.

3.121. Користувач вводить числа a і b . Програма повинна обчислити суму останніх цифр всіх чисел, які належать проміжку $[a;b]$.

3.122. Написати програму, яка виводить у зворотному порядку всі числа, кратні x , на проміжку від 200 до 100. Число x задається користувачем.

3.123. Написати програму, яка виводить у зворотному порядку всі двозначні числа, кратні t . Число t задається користувачем.

3.124. Користувач вводить межі діапазону $[a;b]$. Програма повинна вивести на екран усі числа з цього діапазону, парні числа при цьому збільшивши вдвічі.

3.125. Написати програму, яка виводить на екран усі парні числа в діапазоні $[20; x]$. Число x вводиться з клавіатури. Оператор умови `if` не використовувати.

3.126. Написати програму, яка у зворотному порядку виводить на екран усі числа, кратні 3, у порядку спадання.

3.127. Користувач вводить межі діапазону $[a;b]$. Програма повинна вивести у стовпчик добутки чисел з цього діапазону на випадкове число у діапазоні $[3;20]$.

3.128. Написати програму, яка виводить таблицю множення для числа 11 у діапазоні $[11;20]$.

3.129. Написати програму, яка виводить квадрати і куби чисел від a до b у три стовпчики: саме число, квадрат числа, куб числа.

3.130. Користувач вводить вантажопідйомність автомобілів. Введення даних закінчується введенням числа 0. Програма повинна обчислити сумарну і середню вантажопідйомність автомобілів.

3.131. Написати програму для обчислення найбільшого спільного дільника 2 двозначних чисел, введених з клавіатури.

3.132. Написати програму для обчислення найбільшого спільного дільника 3 двозначних чисел, введених з клавіатури.

3.133. Користувач вводить відстані від початку координат певної кількості точок на координатній прямій. Програма повинна вивести найменше і найбільше значення координат. Введення даних закінчується, коли користувач ввів 0.

3.134. Написати програму, яка обчислює суму відстаней від початку координат усіх точок, координати яких x – цілі числа в діапазоні від 1 до n . Цикл з параметром не використовувати.

3.135. Користувач вводить число Z . Програма повинна вивести на екран найбільше ціле число X у діапазоні $[1; Z]$, яке ділиться на 17. Цикл з параметром не використовувати.

3.136. Користувач вводить ціле число n . Програма повинна вивести суму n чисел в діапазоні від 1 до n , які діляться на 11. Цикл з параметром не використовувати.

3.137. Написати програму, яка виведе на екран усі натуральні числа, квадрат яких не перевищує F . Число F вводиться з клавіатури.

3.138. Написати програму, яка перевіряє, чи діляться введені цілі числа на x без остачі. Введення чисел припиняється після введення числа 0.

3.139. Написати програму, яка виводить на екран куби всіх натуральних чисел, менші від W . Число W вводиться з клавіатури.

3.140. Користувач задає два числа a і b . Програма повинна вивести квадрати всіх чисел, які діляться на 3, в діапазоні $[a;b]$. Цикл з параметром не використовувати.

3.141. Написати програму, яка протабулює функцію $y=x^2+3$ на інтервалі $[a;b]$ з кроком $0,1$. Числа a і b задає користувач.

3.142. Написати програму, яка виводить на екран усі дільники натурального числа X . Число X задається з клавіатури.

3.143. Користувач вводить натуральне число. Програма повинна обчислити суму його цифр.

3.144. Користувач вводить натуральне число. Програма повинна обчислити добуток його цифр.

3.145. Написати програму, яка визначає кількість цифр у натуральному числі, введеному користувачем.

3.146. Написати програму, яка визначає кількість парних цифр у натуральному числі, введеному користувачем.

3.147. Написати програму, яка визначає кількість непарних цифр у натуральному числі, введеному користувачем.

3.148. Користувач вводить натуральне число. Написати програму, яка обчислить суму квадратів його цифр.

3.149. Користувач вводить натуральне число. Написати програму, яка обчислить суму кубів його цифр.

3.150. Користувач вводить натуральне число. Програма повинна вивести на екран найменший дільник s цього числа, причому $s > 1$.

3.151. Написати програму, яка виводить у стовпчик усі двозначні числа, кратні 7. Оператор умови (if) не використовувати.

3.152. Написати програму, яка виводить на екран перші 6 натуральних чисел, більших за 200, які діляться на 23 без остачі.

3.153. У банк поклали суму грошей X грн під процент ($k\%$ річних). Написати програму, яка визначить кількість років, після яких вклад стане Y грн. Цикл з параметром не використовувати. Числа X , Y , k вводяться з клавіатури.

3.154. У банк поклали суму X грн під $R\%$ річних. Програма повинна визначити суму на рахунку через 5 років. Цикл з параметром не використовувати. Числа X і R вводяться з клавіатури.

3.155. Користувач вводить натуральне число. Програма повинна визначити, скільки разів у запису числа зустрічається цифра 7.

3.156. Користувач вводить довжину відрізка AB (натуральне число). Програма повинна визначити найбільшу кількість однакових відрізків (довжини яких є натуральними числами), на які можна розрізати відрізок AB .

3.157. Користувач вводить натуральне число. Програма повинна вивести його у зворотньому порядку.

3.158. Користувач вводить натуральне число N . Програма повинна визначити, чи є у запису цього числа цифри 3 і 7.

3.159. Користувач вводить натуральне число, всі цифри якого різні. Програма повинна вивести найбільшу цифру, яка є у запису числа.

3.160. Користувач вводить натуральне число, всі цифри якого різні. Програма повинна вивести номер позиції найменшої цифри, яка є у запису числа.

3.161. Користувач вводить число N . Програма повинна визначити, чи є це число степенем числа 7.

3.162. Користувач вводить число N . Програма повинна визначити, чи є це число степенем числа 11.

3.163. Користувач вводить натуральне число. Програма повинна визначити, чи повторюється в ньому остання цифра.

3.164. Програма підраховує суму натуральних чисел, починаючи з числа a . Користувач задає початкову межу діапазону a . Програма повинна визначити праву межу діапазону, коли сума всіх чисел досягне 10000.

3.165. Програма підраховує суму натуральних чисел, кратних a , починаючи з числа 255. Число a задається користувачем. Програма повинна визначити кількість чисел, коли сума досягне 5600.

3.166. На числовій прямій задано відрізок довжиною x (задається користувачем). Далі цей відрізок ділиться навпіл, права частина відкидається. Далі – знову ділиться навпіл, і до тих пір, поки довжина відрізка не стане меншою від 1. Програма повинна обчислити кількість поділів.

3.167. Написати програму, яка виведе таблицю значень функції $y=x^2-x+2$ в діапазоні від x_1 до x_2 з кроком 0,1. Числа x_1 і x_2 задаються користувачем.

3.168. Довжина кроку учня дорівнює 40 см. Вийшовши з дому, він робить кроки так, що довжина кожного наступного кроку на 1 см більша за довжину попереднього. За Z м від дому знаходиться школа. Програма повинна вирахувати кількість кроків, які необхідно зробити учню, щоб потрапити до школи.

3.169. На першій гілці дерева сидить одна пташка, на другій – дві і т.д. Всього на дереві сидить Y птахів. Написати програму, яка обчислить кількість гілок дерева.

3.170. Користувач задає число x . Програма повинна вивести на екран найбільше число y , кратне x , менше від 3000.

3.171. Користувач задає число F . Програма повинна вивести на екран найменше число s , кратне 19, причому $s > F$.

3.172. Користувач задає число x . Програма повинна вивести усі його дільники. Цикл з параметром не використовувати.

3.173. Розробити програму, яка виводить всі числа, кратні x , не більші від z . Числа x і z вводяться з клавіатури.

3.174. Написати програму, яка виводить на екран усі натуральні числа n – знаменники дробу $18/n$, при яких дріб буде неправильним.

3.175. Написати програму, яка виводить на екран усі числа a – чисельники дробу $a/22$, при яких дріб буде правильним.

3.176. Написати програму, яка виводить на екран таблицю значень функції $y=x^4-x^3$ від x_1 до x_2 з кроком 0,2. Числа x_1 і x_2 задаються з клавіатури.

3.177. Написати програму, яка виводить на екран таблицю значень для 2 функцій: $y=x^4$ і $y=x^2$ у діапазоні від x_1 до x_2 з кроком 0,1. Межі діапазону x_1 і x_2 задаються користувачем.

3.178. Написати програму, яка обчислює таблицю значень функції $y=x^3+x^2-2x+6$ у діапазоні від x_1 до x_2 з кроком 0,5. Числа x_1 і x_2 задаються користувачем.

3.179. Відрізок довжиною x см ділиться на дві рівні частини, половина відкидається, інша половина ділиться на два і так далі – до тих пір поки частка від ділення не стане меншою за 1. Програма повинна обчислити кількість таких кроків.

3.180. Написати програму, яка обчислює суму парних чисел, введених з клавіатури. Введення даних закінчується числом 0.

3.181. Написати програму, яка обчислює суму значень функції $y=x^2$ у діапазоні від x_1 до x_2 з кроком 0,4. Цикл з лічильником не використовувати.

3.182. Написати програму, яка виводить таблицю значень функції $y=0.3x+z$, де z – випадкове число у діапазоні $[0;15]$ з кроком 0,1. Цикл з параметром не використовувати.

3.183. Написати програму, яка обчислює суму квадратів натуральних чисел, введених користувачем. Зчитування даних з екрану повинні припинятися у таких випадках: 1) користувач ввів 0; 2) сума перевищила 990000.

3.184. Під час нагрівання води у паровому казані її температура збільшувалася від 0° до 100°C , причому кожної наступної хвилини температура зростала на величину 1°C швидше, ніж за попередню. Написати програму, яка визначить час, потрібний для закипання води у казані.

3.185. Написати програму, яка виводить суму двоцифрових чисел, кратних 3. Цикл з лічильником не використовувати.

3.186. Користувач задає натуральне число Y – межу діапазону випадкових чисел. Програма повинна обчислювати суму випадкових чисел у діапазоні $[0; Y]$ до тих пір, поки вона не перевищить 1000, і підрахувати кількість цих чисел.

3.187. Вантажопідйомність автомобілів являє собою випадкове число у діапазоні $[0;5]$ (т). Програма повинна визначити

мінімальну кількість автомобілів, які необхідні для перевезення вантажу Z т. Число Z вводить користувач.

3.188. Користувач задає натуральне число N . Написати програму, яка генеруватиме випадкові числа у діапазоні $[0;20]$ до тих пір, поки число 17 не буде згенероване N разів. Програма повинна вивести на екран загальну кількість згенерованих чисел.

3.189. Користувач задає натуральне число N . Програма повинна генерувати випадкові числа у діапазоні від -10 до 10 до тих пір, поки кількість додатних згенерованих чисел на досягне N . Програма повинна вивести на екран кількість запусків генератора випадкових чисел.

3.190. Програма повинна згенерувати N від'ємних чисел. Проте генератор випадкових чисел налаштований так, щоб генерувати випадкові числа у діапазоні $[-5;10]$. Програма повинна вивести на екран загальну кількість згенерованих чисел, створених до тих пір, поки кількість від'ємних чисел не досягла N . Число N вводиться з клавіатури.

3.191. Користувач задає число A . Програма повинна генерувати випадкові числа у діапазоні $[-A;A]$ до тих пір, поки кількість згенерованих від'ємних чисел не перевищить кількість згенерованих додатних чисел, і підрахувати загальну кількість згенерованих чисел.

3.192. Написати програму, яка обчислює суму трицифрових чисел, кртних 29. Цикл з лічильником не використовувати.

3.193. Хлопчик підкидає кубик, на гранях якого написані числа від 1 до 6. Програма повинна підрахувати кількість підкидань кубиків, виконаних до того, як число 6 випало 6 разів.

3.194. Хлопчик підкидає кубик, на гранях якого написані числа від 1 до 6, до тих пір, поки кількість парних чисел, що випала, досягне 25. Програма повинна вивести на екран кількість підкидань кубика.

3.195. На колесі рулетки написані числа від 1 до 50. Гравець поставив гроші на непарне число. Програма повинна підрахувати кількість пусків колеса рулетки до тих пір, поки випаде виграш.

3.196. Написати програму, яка обчислюватиме суму двоцифрових чисел, введених користувачем. Введення даних припиняється, коли сума досягне або перевищить 100. Програма повинна вивести на екран суму і кількість чисел.

3.197. Гоночний автомобіль стартує з місця, розганяючись до певної швидкості так, що кожної секунди його швидкість зростає на 8 м/с. Написати програму, яка обчислить відстань, яка пройде автомобіль, розігнавшись до швидкості v м/с. Число v вводиться з клавіатури.

3.198. Мандрівники на мотоциклах проїжджали кожного дня x км. Написати програму, яка виведе кількість днів, потрібних для подолання мандрівниками z км. Число x є випадковим у діапазоні $[200;400]$, а число z задається користувачем.

3.199. Велосипедист рухається з місця, розганяючись на $0,5$ м/с за 1 с. Програма повинна вивести відстань, після проходження якої велосипедист розганяється до швидкості v м/с.

3.200. Користувач загадує число x у діапазоні $[1;50]$. Програма повинна обчислити кількість пусків рулетки, поки на ній не випаде число x .

Розділ 4. Комп'ютерна графіка в Python

Для роботи з комп'ютерної графікою призначений модуль turtle. Його основні команди наведено в табл. 1.4.

Таблиця 4.1

Функції модуля turtle

Назва функції	Опис	Приклад використання
<code>turtle.forward(x)</code>	Рух вперед на x пікселів	<code>turtle.forward(50)</code>
<code>turtle.backward(x)</code>	Рух назад на x пікселів	<code>turtle.backward(20)</code>
<code>turtle.left(z)</code>	Повернути вліво на z градусів	<code>turtle.left(90)</code>
<code>turtle.right(z)</code>	Повернути вправо на z градусів	<code>turtle.right(90)</code>
<code>turtle.penup()</code>	Не залишати сліду під час руху	<code>turtle.penup()</code>
<code>turtle.circle(r)</code>	Намалювати коло радіусом r	<code>turtle.circle(50)</code>
<code>turtle.shape('x')</code>	Встановити значок виконавця (arrow, turtle, circle, square, triangle, classic)	<code>turtle.shape('circle')</code>
<code>turtle.stamp()</code>	Скопіювати об'єкт Виконавець у поточному місці	<code>turtle.stamp()</code>
<code>turtle.color('red')</code>	Встановити колір	<code>turtle.color('red')</code>
<code>turtle.begin_fill()</code>	Викликається перед малюванням фігури, яку слід зафарбувати	<code>turtle.begin_fill()</code>
<code>turtle.end_fill()</code>	Викликається після завершення малювання фігури	<code>turtle.end_fill()</code>
<code>turtle.width()</code>	Встановити товщину лінії в пікселях	<code>turtle.width(10)</code>
<code>turtle.goto(x,y)</code>	Перемістити виконавця у точку з координатами (x, y)	<code>turtle.goto(100,150)</code>
<code>turtle.dot(x)</code>	Малювання точки діаметром x пікселів	<code>turtle.dot(5)</code>
<code>turtle.bgcolor('color')</code>	Встановити колір тла	<code>turtle.bgcolor('white')</code>

turtle.title ('Заголовок')	Встановити заголовок вікна	turtle.title ('Моя програма')
turtle.shapesize(1,2,3)	Зміна розмірів виконавця	turtle.shapesize(1,2,3)
turtle.pensize(x)	Встановити товщину ліній x пікселів	turtle.pensize(3)
turtle.fillcolor('color')	Встановити колір тла об'єкта color	turtle.fillcolor('black')
turtle.pencolor('color')	Встановити колір ліній color	turtle.pencolor('green')
turtle.speed(x)	Встановити швидкість руху пензля x	turtle.speed(5)
turtle.undo()	Відміна останніх дій	turtle.undo()
turtle.clear()	Очищення робочої області	turtle.clear()
turtle.pendown()	Знову почати малювання	turtle.pendown()
turtle.clone()	Клонувати об'єкт	player_two = player_one.clone()
turtle.Turtle()	Присвоїти об'єкт змінній	player_one = turtle.Turtle()
turtle.screen().screensize()	Повернути розміри екрану	turtle.screen().screensize()
turtle.screen.setup(x,y)	Встановити розміри екрану	turtle.screen.setup(x,y)

Приклад 4.1. Написати програму, яка малює довільну фігуру.

Розв'язання:

```
import turtle
turtle.shape('arrow')
turtle.color('red')
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.color('green')
turtle.forward(100)
turtle.right(90)
turtle.color('blue')
turtle.forward(100)
```

Приклад 4.2. Написати програму, яка малює квадрат, змінюючи колір і товщину ліній.

Розв'язання

```
import turtle
d=int(input("Введіть розміри квадрата"))
turtle.color("red")
turtle.width(4)
turtle.forward(d)
turtle.color("green")
turtle.width(5)
turtle.left(90)
turtle.forward(d)
turtle.color("blue")
turtle.width(2)
turtle.left(90)
turtle.forward(d)
turtle.color("yellow")
turtle.width(7)
turtle.left(90)
turtle.forward(d)
turtle.color("red")
turtle.width(3)
turtle.left(90)
```

Приклад 4.3. Написати програму, яка малює квадрат. Використати оператор циклу.

```
import turtle
#d=int(input("Введіть розміри квадрата"))
d=200
turtle.color('white')
turtle.goto(-100,-100)
turtle.color("red")
w=3
turtle.width(w)
for i in range (20):
    w=1+w
    turtle.width(w)
    d=d+7
    turtle.forward(d)
    turtle.left(90)
```

Приклад 4.4. Написати програму, яка малює коло синього кольору з радіусом, що дорівнює випадковому числу в межах від 150 до 200.

```
import random
import turtle
d=random.randint(150,200)
turtle.color('white')
turtle.goto(0,-d)
turtle.shape('arrow')
turtle.color('blue')
turtle.circle(d)
```

Приклад 4.5. Написати програму, яка малює квадрат із сторонами різного кольору і з різною швидкістю. Оператор циклу не використовувати.

```
import turtle
d=3
v=1
turtle.penup()
turtle.goto(0,-100)
turtle.pendown()
turtle.width(d)
turtle.speed(v)
turtle.color('red')
turtle.forward(100)
turtle.left(90)
d=d+2
v=v+4
turtle.width(d)
turtle.speed(v)
turtle.color('blue')
turtle.forward(100)
turtle.left(90)
d=d+2
v=v+4
turtle.width(d)
turtle.speed(v)
turtle.color('maroon')
turtle.forward(100)
turtle.left(90)
d=d+2
v=v+4
turtle.width(d)
turtle.speed(v)
turtle.color('gray')
turtle.forward(100)
turtle.left(90)
```

Приклад 4.6. На зображенні показан поворот черепашки на 20 градусів. Ты же можешь попробовать, например, 30 или 40.

```
turtle.left(20)

turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)

turtle.left(30)

turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)

turtle.left(40)

turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
turtle.forward(50)
turtle.left(90)
```

Приклад 4.7. Написати програму, яка встановлює розміри вікна 800x600 пікселів і малює осі координат.

```
import turtle
t=turtle.Turtle()
t.screen.setup(800,600)
t.penup()
t.goto(0,300)
t.pendown()
```

```
t.goto(0,-300)
t.penup()
t.goto(-400,0)
t.pendown()
t.goto(400,0)
```

Завдання для виконання

4.1. Написати програму, яка малює три кола радіусами 30, 50 і 70 пікселів. Колір ліній – зелений, синій і жовтий.

4.2. Написати програму, яка малює на екрані прямокутник розмірами 70x180 пікселів. Колір ліній – жовтий, колір заливки фігури – синій.

4.3. Написати програму, яка малює на екрані три горизонтальні паралельні лінії різною товщиною. Колір ліній відповідно синій, зелений, червоний. Лінії повинні починатися від лівого краю і закінчуватися на правому краю.

4.4. Написати програму, яка малює на екрані дві чорні вертикальні лінії товщиною 6 пікселів за допомогою фігур-черепах. У програмі використати клонування об'єктів.

4.5. Написати програму, яка малює на екрані дві діагоналі жовтого і зеленого кольору. Товщина ліній – відповідно 8 і 12 пікселів.

4.6. Написати програму, яка малює на екрані квадрат. Швидкість малювання, товщину ліній і їх колір зробити різними для кожної сторони квадрата.

4.7. Написати програму, яка малює рамку по периметру вікна turtle.

4.8. Написати програму, яка змінює фігуру виконавця з черепахи на стрілку і навпаки з періодичністю 6 с. Використати цикл for.

4.9. Написати програму, яка змінює розміри фігури (черепахи) від 2 до 8.

4.10. Написати програму, яка малює правильний шестикутник зі стороною 120 пікселів. Тип виконавця – черепаха, колір черепахи – зелений, колір ліній – синій.

4.11. Написати програму, яка малює на формі велику літеру Т, причому вертикальна і горизонтальна лінії повинні бути різного кольору.

4.12. Написати програму, яка малює на формі велику літеру Ш, кожна лінія якої має різний колір. Доопрацювати програму таким чином, щоб черепаха виконувала якомога менше рухів.

4.13. Написати програму, яка малює на формі велику літеру Z довільних розмірів. Усі лінії повинні бути різної товщини і різного кольору.

4.14. Користувач вводить довжини сторін прямокутника і кут нахилу до осі x . Виконавець повинен намалювати прямокутник заданих розмірів і під заданим кутом.

4.15. Користувач вводить сторону квадрата. Програма повинна повернути квадрат на 45° так, щоб став ромбом, і накреслити діагоналі квадрата.

4.16. Користувач вводить сторону шестикутника і координати startової точки. Програма повинна намалювати шестикутник з початком у startовій точці.

4.17. Написати програму, яка малює пісочний годинник довільних розмірів.

4.18. Написати програму, яка малює велику літеру Ф довільних розмірів.

4.19. Користувач задає координати двох сусідніх верхніх точок прямокутника і його висоту. Програма повинна намалювати прямокутник.

4.20. З клавіатури вводяться довжини двох діагоналей ромба, які паралельні осям координат, а центр ромба співпадає з центром координат. Програма повинна намалювати ромб.

4.21. Написати програму, яка малює рівносторонній трикутник з довжиною сторони, введеною з клавіатури.

4.22. Написати програму, яка малює рівнобедрений трикутник, основа якого паралельна осі x . довжини сторін вводить користувач.

4.23. Написати програму, яка малює цифру 8 з довільними розмірами.

4.24. З клавіатури вводяться розміри рівнобедреного трикутника. Програма повинна намалювати трикутник з основою вгорі, основа повинна бути паралельна осі x .

4.25. З клавіатури вводяться основи і бічні сторони рівнобічної трапеції. Програма повинна намалювати трапецію з основами, паралельними осі x .

4.26. Написати програму, яка малює велику літеру H довільних розмірів.

4.27. Написати програму, яка малює велику літеру Y довільних розмірів.

4.28. Написати програму, яка запитує у користувача координати точки і малює її на декартовій площині. Крім того, програма повинна побудувати осі декартової площини.

4.29. Користувач вводить довжини основ і сторони прямокутної трапеції, кути при якій дорівнюють 90° . Програма повинна намалювати трапецію.

4.30. Написати програму, яка малює зелену крапку у точці з координатами, заданими користувачем.

4.31. Користувач задає модулі координат точки $|x|$ і $|y|$. Програма повинна намалювати чотири крапки різного кольору в точках з координатами $(x;y)$, $(-x;y)$, $(x;-y)$, $(-x;-y)$.

4.32. Користувач задає дві сторони паралелограма і кут між ними. Програма повинна намалювати паралелограм так, щоб його основа була паралельна осі x .

4.33. Користувач задає дві сторони паралелограма і кут між ними. Програма повинна намалювати паралелограм так, щоб його основа була паралельна осі y .

4.34. Написати програму, яка малює рівносторонній трикутник із стороною, заданою користувачем. Стартову точку також задає користувач.

4.35. Написати програму, яка малює дві діагоналі екрану суцільною лінією товщиною 3.

4.36. Написати програму, яка малює прямокутник з розмірами 300×200 , а потім проводить діагоналі цього прямокутника.

4.37. Написати програму, яка малює ромб з стороною 300 і гострим кутом 30° .

4.38. Користувач задає дві сторони трикутника і кут між ними в градусах. Програма повинна намалювати цей трикутник.

4.39. Написати програму, яка малює три горизонтальні лінії від лівої межі вікна до правої, причому кольори цих ліній зеленій, червоний, синій, а товщина ліній 2, 4 і 6 пікселів.

4.40. Написати програму, яка малює два квадрати: один всередині іншого. Розміри внутрішнього квадрата задає користувач, розміри зовнішнього квадрата на 2 пікселів більші, ніж внутрішнього. Товщина ліній зовнішнього квадрата 3, а внутрішнього – 1. Колір ліній зовнішнього квадрата зеленій, внутрішнього квадрата – коричневий.

4.41. Написати програму, яка малює 4 квадрата з розмірами, заданими користувачем, так, що вони утворюють один великий квадрат. Колір ліній – синій, тип ліній – пунктир.

4.42. Написати програму, яка малює зелене коло на жовтому тлі.

4.43. Написати програму, яка малює три квадрати, довжина сторін яких зменшується на 20 пікселів. Початкову довжину сторони квадрата задає користувач. Колір ліній – синій.

4.44. Написати програму, яка малює чотири рівносторонніх трикутники однакових розмірів, кожен наступний буде зсунуто вправо відносно попереднього на 20 пікселів.

4.45. Написати програму, яка малює на екрані рівносторонній трикутник з довжиною сторони, заданою користувачем. Кольори сторін: зелений, синій, чорний.

4.46. Розробити програму, яка малює на екрані літеру W з довжиною лінії, заданою користувачем.

4.47. Написати програму, яка малює паралелограм, дві сторони якого горизонтальні. Довжини сторін вводить користувач. Кольори сторін: жовтий, зелений, синій, червоний.

4.48. Розробити програму, яка малює три квадрати однакових розмірів, зафарбованих синім, жовтим і зеленим кольорами.

4.49. Написати програму, яка схематично малює зображення пісочного годинника. Розміри основ пісочного годинника задає користувач.

4.50. Написати програму, яка малює рядок квадратних клітинок зі стороною 50 пікселів і зеленим кольором ліній. Кількість клітинок задає користувач.

4.51. Написати програму, яка намалює N квадратів з спільним центром. Сторона кожного наступного квадрата більша за сторону попереднього на 10 пікселів. Початкові розміри квадрата 30 пікселів.

4.52. Написати програму, яка намалює N кіл з спільним центром. Радіус кожного наступного кола на 20 пікселів більший за радіус попереднього. Кількість кіл і радіус першого кола задає користувач.

4.53. Написати програму, яка намалює N правильних трикутників із стороною a так, щоб їх основи утворювали собою горизонтальну лінію. Величини N , a вводяться з клавіатури.

4.54. Написати програму, яка намалює квадрати з стороною a по периметру вікна. Сторона квадрата задається з клавіатури.

4.55. Написати програму, яка малює рамку по периметру вікна лініями товщиною 5 пікселів синім, зеленим і білим кольорами.

4.56. Прямокутний паралелепіпед має розміри a, b, c (задаються користувачем). Програма повинна побудувати на екрані розгортку цього паралелепіпеда.

4.57. Довжина ребра куба a (задається користувачем). Написати програму, яка побудує розгортку куба на екрані.

4.58. Користувач задає діаметр і висоту циліндра (в пікселях). Програма повинна побудувати розгортку циліндра на екрані.

4.59. В основі прямої піраміди лежить квадрат зі стороною a . Висота піраміди h . Написати програму, яка намалює розгортку піраміди на екрані.

4.60. Користувач задає розміри прямокутного паралелепіпеда a, b, c . Написати програму, яка намалює три проекції цього паралелепіпеда.

4.61. Написати програму, яка ділить коло радіусом R на 8 рівних секторів. Значення R задає користувач.

4.62. Користувач задає довжину основи рівнобедреного трикутника і його висоту. Програма повинна намалювати трикутник на екрані.

4.63. Написати програму, в якій екран буде заштриховано вертикальними червоними лініями товщиною 2 пікселя. Відстань між лініями штриховки задає користувач.

4.64. Розробити програму, у якій виконавець у вигляді стрілки заштриховує екран горизонтальними зеленими лініями. Відстань між лініями задається користувачем (у межах від 30 до 50 пікселів).

4.65. Створити програму, яка заштриховує прямокутник розмірами 200×400 пікселів похилими синіми лініями товщиною 2 пікселя під кутом 45° .

4.66. Написати програму, яка малює ромб з діагоналями d_1 і d_2 (задаються з клавіатури), причому довша діагональ знаходиться під кутом 30° до вертикалі.

4.67. Написати програму, яка малює два квадрати з стороною a , основи яких нахилені під кутом 25° до осі x . Величину a вводить користувач.

4.68. Користувач задає кут нахилу прямокутника до осі x . Програма повинна намалювати прямокутник зі сторонами 40 і 100 пікселів під цим кутом.

4.69. Написати програму, яка малює два кола радіусом R так, щоб їх осі знаходилися на одній горизонтальній лінії, а самі кола дотикалися. Радіус R задається користувачем.

4.70. Користувач задає довжину сторони a рівностороннього трикутника. Програма повинна намалювати трикутник так, щоб його основа була горизонтальною, і середню лінію трикутника, паралельну до цієї основи.

4.71. З клавіатури задають довжину сторони рівностороннього трикутника. Програма повинна намалювати цей трикутник, а також три середні лінії трикутника. Основа трикутника повинна бути горизонтальною.

4.72. Написати програму, яка малює паралелограм з сторонами заданої довжини так, щою його основа була горизонтальною. Далі програма повинна провести середню лінію паралелограма, паралельну осі x .

4.73. Написати програму, яка малює рівнобічну трапецію так, щоб її основи були паралельні осі x , а також середню лінію трапеції.

4.74. Написати програму, яка малює рівнобедрений трикутник так, щоб його основа була вертикальною, а також середню лінію трикутника, паралельну основі. Сторони трикутника задаються з клавіатури.

4.75. Розробити програму, яка малює рівнобедрений прямокутний трикутник так, щоб його основа і середня лінія були паралельні осі x . З клавіатури задається катет трикутника.

4.76. Написати програму, яка малює трапецію так, щоб її основи були горизонтальними. Користувач задає довжини основ і кути при нижній основі трапеції.

4.77. Написати програму, яка малює трикутник так, щоб його основа була паралельною осі x , і середню лінію цього трикутника. Користувач задає довжину основи і кути при основі.

4.78. Написати програму, яка малює рівнобічну трапецію так, щоб її основи були паралельні осі OY . Користувач задає довжини основ і кут при більшій основі.

4.79. Написати програму, яка зображає перевернутий рівнобедрений трикутник так, щоб його основа і середня лінія були паралельні осі OX . Користувач задає довжину основи і кут, протилежний основі.

4.80. Користувач задає довжини сторін паралелограма і кут між ними. Програма повинна намалювати паралелограм так, щоб його основи були паралельні осі OY .

4.81. З клавіатури задаються довжина сторони ромба і кут між сторонами. Програма повинна намалювати ромб так, щоб дві його сторони були паралельними осі OY .

4.82. Написати програму, яка зображує рівнобедрений трикутник, а всередині його – подібний йому трикутник менших розмірів. Розміри обох трикутників встановити довільно.

4.83. Написати програму, яка зображує прямокутну трапецію довільних розмірів так, щоб її основи були паралельні осі OX .

4.84. Розробити програму, яка малює прямокутну трапецію з довжинами основ a і b , висотою h так, щоб основи були паралельні осі OY .

4.85. Розробити програму, яка малюватиме прямокутну трапецію так, що її основи розміщуються вертикально. З клавіатури задаються дві основи і менша бічна сторона трапеції.

4.86. Розробити програму, яка малюватиме прямокутну трапецію так, що її основи розміщуються горизонтально. З клавіатури задаються дві основи і менша бічна сторона трапеції.

4.87. Написати програму, яка малює на екрані умовне позначення резистора. Розміри довільні.

4.88. Написати програму, яка малюватиме умовне позначення лампочки. Розміри довільні.

4.89. Написати програму, яка малює правильний шестикутник довільних розмірів так, що товщина лінії, яка обмежує фігуру, збільшується кожного разу за 2 пікселі, а колір змінюється.

4.90. Користувач задає радіус кола R і натуральне число N ($N < 12$). Розробити програму, яка малює коло і поділяє його на N ривних частин.

4.91. Написати програму, яка малює правильний шестикутник довільних розмірів і розбиває його на шість однакових трикутників. Кольори ліній: зелений, синій, жовтий, червоний, чорний, коричневий.

4.92. Написати програму, яка розбиває квадрат довільних розмірів на чотири однакових квадрата. Всі лінії повинні бути різних кольорів.

4.93. Написати програму, яка розбиває прямокутник довільних розмірів на чотири однакових прямокутника.

4.94. Користувач задає довжину і ширину прямокутника. Програма повинна розбити прямокутник на чотири однакових прямокутника.

4.95. Написати програму, яка заштриховує вікно вертикальними лініями зеленого кольору товщиною 2 пікселя. Відстань між лініями – 35 пікселів.

4.96. Написати програму, яка заштриховує вікно горизонтальними синіми лініями товщиною 4 пікселі. Відстань між лініями – 20 пікселів.

4.97. Написати програму, яка малює у вікні ромб так, щоб вершини ромба лежали на серединах сторін вікна. Параметри вікна зчитати з налаштувань програми.

4.98. Написати програму, яка малює у вікні шестикутник так, щоб дві його основи лежали на верхній і нижній межах вікна, а права і ліва вершини знаходилися на серединах бокових сторін вікна.

4.99. Написати програму, яка малює восьмикутник так, щоб його верхня і нижня основи лежали на верхній і нижній сторонах вікна, а ліва і права сторони восьмикутника – на бічних сторонах вікна. Координати вершин восьмикутника повинні бути прив'язаними до розмірів вікна.

4.100. Написати програму, яка малює на екрані три прямокутника однакових розмірів, розташованих в один горизонтальний ряд. Розміри прямокутників повинні бути прив'язані до розмірів вікна.

Розділ 5. Алгоритми обробки рядкових і символьних величин

Часто виникає необхідність створити програму для роботи з текстом. Символьний тип – це один знак: літера, цифра або інший знак. Рядок – це набір символів. У рядку їх може бути два, а може бути 127.

Текстові величини набувають значень під час ініціалізації або після зчитування з клавіатури.

Значення символьних і рядкових величин поміщаються в одинарні або подвійні лапки.

Деякі послідовності символів є зарезервованими.

Таблиця 5.1

Escape-послідовності

Escape-послідовність	Характеристика
\t	Табуляція
\n	Перехід на новий рядок
\"	Подвійні лапки
'	Одинарні лапки

Символи в рядку мають порядковий номер, який починається з 0.

Над рядковими і символьними величинами виконують різні операції:

- конкатенація (склеювання) – позначається +. З її допомогою об'єднуються два або більше рядків в один;

- вирізання частини рядка – копіювання частини рядка у нову змінну;

- перепис рядка у зворотному порядку.

Копіювання частини рядка:

- від початку рядка до символу з номером n: `ss=gyadok[:n]`;

- від символу з номером a до символу з номером b: `ss=gyadok[a:b]`;

- від символу з номером a до кінця рядка: `ss=gyadok[a:]`.

Для запису рядка у зворотному порядку використовують операцію `ss=gyadok[::-1]`.

Рядок можна подовжити задану кількість разів: `print('OK'*7)`.

У результаті буде виведено повідомлення:

ОКОКОКОКОКОКОКО

Для визначення довжини рядка служить функція `len()`.

Для того, щоб розмістити у рядку символ `\`, необхідно набрати

`\\`. Проте у рядку відобразиться лише один такий символ.

Приклади програм

Приклад 5.1. Написати програму, яка виводить рядок у зворотному порядку. (string01)

```
s1=input("Введіть рядок")
s2=s1[::-1]
print(s2)
```

Приклад 5.2. Користувач вводить рядок, який містить парну або непарну кількість символів. Написати програму, яка виводить один (якщо довжина рядка непарна) або два (якщо довжина рядка парна) середні символи рядка. (string02)

```
s=input("Введіть рядок")
L=len(s)
index1=1
index2=2
s1=' '
s2=' '
if L%2==1:
    index1=L//2
    s1=s[index1]
    print(s1)
else:
    index1=int(L/2-1)
    index2=int(index1+1)
    s1=s[index1]
    s2=s[index2]
    print(s1+s2)
```

Приклад 5.3. Написати програму, яка запитує у користувача окремо ім'я і прізвище, а потім виводить їх як один рядок. (string03)

```
name=input("Введіть ім'я")
surname=input("Введіть прізвище")
probil=" "
s=name+probil+surname
print(s)
```

Приклад 5.4. Користувач вводить два імені дітей. Написати програму, яка зчитує їх, а потім міняє місцями. (string04)

```
s1=input("Введіть перше ім'я")
s2=input("Введіть друге ім'я")
s=s1
s1=s2
s2=s
```

Приклад 5.5. Дано рядок. Користувач вводить номер символу у рядку. Програма повинна вивести цей символ. (string05)

```
import math
x='Привітання зі святом'
```

```
i=int(input("Введіть номер символу: "))
L=len(x)
if i<L:
    print(x[i])
else:
    print("введений номер більший за довжину рядка")
```

Завдання для виконання

5.1. Написати програму, яка запитує у користувача назву країни і її столицю, після чого виводить всі дані в один рядок. Наприклад: «Столиця країни *Україна* місто *Київ*». Курсором виділені ті дані, які ввів користувач.

5.2. Користувач вводить назву футбольного клубу. Написати програму, яка визначає кількість літер у назві.

5.3. Користувач вводить назви двох міст. Програма повинна вивести назву, яка містить більше символів.

5.4. Користувач вводить прізвище. Програма повинна вивести кількість символів у ньому.

5.5. Користувач вводить назви трьох міст. Програма повинна вивести найдовшу і найкоротшу назви.

5.6. Користувач вводить два імені дітей. Написати програму, яка зчитує їх, а потім міняє місцями.

5.7. Користувач вводить слово. Програма повинна вивести його третій символ.

5.8. Користувач вводить слово. Програма повинна вивести його останній символ.

5.9. Користувач вводить окремо рядок і номер. Програма повинна вивести на екран символ під цим номером.

5.10. Користувач вводить рядок. Написати програму, яка порівнює другий і п'ятий символи у ньому.

5.11. Користувач вводить слово. Програма повинна визначити, чи починається і закінчується це слово на одну літеру.

5.12. Користувач вводить окремо два слова. Програма повинна визначити, чи починається друге слово на ту саму літеру, на яку закінчується перше.

5.13. Користувач вводить слово. Програма повинна вивести буквосполучення, яке складається з першої, третьої і п'ятої літер.

5.14. Користувач вводить слово. Написати програму, яка виводить буквосполучення, яке складається з першої і останньої літер.

5.15. Користувач вводить слово, яке складається з парної кількості символів. Написати програму, яка виводить на екран першу половину слова.

5.16. Користувач вводить слово. Програма повинна вивести його частину, яка складається з третього, четвертого і п'ятого символів.

5.17. Користувач вводить окремо слово і два числа – a і b . Програма повинна вивести на екран частину слова, яка починається з символу під номером a і закінчується символом під номером b .

5.18. Користувач вводить слово, яке складається з парної кількості літер. Написати програму, яка міняє місцями обидві половини слова.

5.19. Користувач вводить слово і число z . Написати програму, яка вирізає перші z літер і переносить їх у кінець слова.

5.20. Користувач вводить назву міста. Програма повинна вивести всі літери у стовпчик.

5.21. Користувач вводить слово. Написати програму, яка виводить на екран слово у зворотному порядку.

5.22. Користувач вводить слово. Програма повинна створити друге слово, яке складається з непарних літер першого слова.

5.23. Користувач вводить назву міста. Написати програму, яка створює слово, утворене прочитанням назви міста у зворотному напрямку.

5.24. Користувач вводить символ. Написати програму, яка утворює слово, що складається з 7 таких символів.

5.25. Користувач вводить слово. Програма повинна створити рядок, який складається з 3 таких слів.

5.26. Користувач вводить слово. Програма повинна створити рядок, який складається з 4 таких слів, розділених пробілом.

5.27. Користувач задає символ і ціле число N . Програма повинна створити рядок, який складається з N повторів введеного символу.

5.28. Написати програму, яка зчитує слово, дописує на початку слова 6 символів «=», а в кінці – 4 символи «@».

5.29. Написати програму, яка зчитує з екрану слово і дописує на початку слова і в кінці стільки символів «+», скільки літер у слові.

5.30. Користувач вводить окремо два слова різної довжини. Програма повинна обрізати довше слово до розмірів коротшого слова.

5.31. Користувач вводить окремо два слова різної довжини. Програма повинна збільшити коротше слово до розмірів довшого слова, додавши відповідну кількість символів «*».

5.32. Користувач вводить окремо два слова різної довжини. Програма повинна замінити у довшому слові перші символи на коротке слово.

5.33. Написати програму, яка підраховує кількість літер «і» у введеному користувачем слові.

5.34. Написати програму, що визначить, яких літер – «а» чи «б» більше у тексті.

5.35. Написати програму, яка виводить у стовпчик усі голосні літери слова, введеного користувачем.

5.36. Користувач вводить речення. Програма повинна вивести у стовпчик всі символи, які мають парний номер.

5.37. Користувач вводить рядок. Програма повинна вивести всі літери «о» і «а» з цього рядка.

5.38. Користувач вводить рядок. Програма повинна вивести всі буквосполучення «цьк», які є в рядку.

5.39. Користувач вводить рядок. Написати програму, яка виводить на екран усі входження двох окремих символів, заданих користувачем.

5.40. Користувач вводить слово, яке складається з різних літер. Написати програму, яка виводить номер символу «а» у рядку.

5.41. Користувач вводить рядок. Програма повинна вивести 2 частини рядка – з першого по п'ятий і з десятого по п'ятнадцятий символи.

5.42. Користувач вводить речення. Написати програму, яка підраховує кількість літер «ф» у ньому.

5.43. Написати програму, яка підраховує кількість пробілів у рядку, введеному користувачем.

5.44. Користувач вводить окремо рядок і символ. Програма повинна підрахувати кількість таких символів у рядку.

5.45. Написати програму, яка підраховує відсоток символів «у» у рядку, введеному користувачем.

5.46. Написати програму, яка запитує у користувача рядок і вставляє між усіма символами дефіс.

5.47. Написати програму, яка підраховує кількість речень у введеному користувачем рядку.

5.48. Написати програму, яка підраховує загальну кількість символів «+» і «-» у рядку, введеному користувачем.

5.49. Користувач вводить речення. Програма повинна підрахувати кількість голосних літер у ньому.

5.50. Користувач вводить речення. Програма повинна підрахувати кількість приголосних літер у ньому.

5.51. Користувач вводить рядок довільної довжини. Програма повинна визначити, яка з літер – «о» чи «у» зустрічається частіше і наскільки.

5.52. Написати програму, яка визначає, чи є коми в реченні, введеному користувачем.

5.53. Користувач вводить рядок, що складається з слів, розділених пробілами. Програма повинна визначити кількість слів.

5.54. Користувач вводить рядок. Програма повинна визначити, який із символів – «а» чи «о» зустрічається раніше.

5.55. Користувач вводить речення. Написати програму, яка повинна вивести частину речення від початку до першої коми. Якщо кома у реченні відсутня, програма повинна вивести все речення.

5.56. Користувач вводить кілька слів, розділених пробілами. Програма повинна вивести перше слово (пробілів на початку і в кінці рядка немає).

5.57. Користувач вводить кілька слів, розділених пробілами. Написати програму, яка буде виводити останнє слово.

5.58. Користувач вводить речення, яке складається з кількох слів, розділених пробілами. Програма повинна вивести друге слово.

5.59. Користувач вводить речення, яке складається не менше ніж з чотирьох слів, розділених пробілами. Написати програму, яка виводить останні два слова окремо кожне.

5.60. Написати програму, яка визначає: 1) порядковий номер першої з літер «і»; 2) порядковий номер останньої з літер «і».

5.61. Користувач вводить слово. Програма повинна визначити, чи є це слово паліндромом.

5.62. Користувач вводить речення. Програма повинна визначити кількість літер «е» у першому слові.

5.63. Користувач вводить речення. Програма повинна вивести індекс першої пари однакових сусідніх символів.

5.64. Користувач вводить речення. Програма повинна визначити індекс першого входження буквосполучення «ка».

5.65. Користувач вводить речення. Програма повинна вивести частину речення, яка знаходиться між першою і другою комами. Якщо другої коми немає, програма повинна вивести частину речення від коми до кінця рядка.

5.66. Користувач вводить рядок. Програма повинна замінити всі символи «а» на символи «у».

5.67. Написати програму, яка міняє місцями другу і шосту літери у рядку, введеному користувачем.

5.68. Користувач вводить слово, довжина якого становить парну кількість символів. Програма повинна поміняти місцями парні і непарні літери.

5.69. Користувач вводить рядок з 8 символів. Програма повинна поміняти у зворотному порядку всі символи з індексами від 3 до 6.

5.70. Користувач вводить слово. Програма повинна видалити з нього третю літеру.

5.71. Користувач вводить слово. Програма повинна видалити з нього літеру під номером N. Число N також вводиться користувачем.

5.72. Користувач вводить слово. Програма повинна видалити з нього першу літеру «о», якщо вона є.

5.73. Користувач вводить слово. Програма повинна видалити з слова останню літеру «у», якщо вона є.

5.74. Користувач вводить слово. Якщо у ньому непарна кількість символів, то програма повинна видалити з нього середню літеру. Якщо у слові парна кількість символів, то програма повинна видалити дві середні літери.

5.75. Користувач вводить рядок. Програма повинна видалити всі символи з n-го до k-го ($n < k$).

5.76. Написати програму, яка видаляє з рядка перший пробіл.

5.77. Написати програму, яка видаляє з рядка останній пробіл.

5.78. Написати програму, яка видаляє з рядка всі символи, які стоять на непарних місцях.

5.79. Користувач вводить символ. Написати програму, яка визначає, чи є цей символ цифрою.

5.80. Користувач вводить рядок. Програма повинна вивести у стовпчик всі символи, які є цифрами.

5.81. Написати програму, яка визначає кількість цифр у рядку, введеному користувачем.

5.82. Користувач вводить рядок. Програма повинна знайти суму всіх цифр у рядку.

5.83. Користувач вводить рядок. Програма повинна вивести найбільшу цифру у рядку.

5.84. Користувач вводить рядок. Програма повинна визначити номер першого входження цифри 3.

5.85. Користувач вводить рядок, у якому є кілька цифр. Програма повинна вивести число, утворене цими цифрами, записаними у порядку їх входження у рядок.

5.86. Написати програму, яка визначає кількість цифр у дійсному числі, якщо дробова частина відокремлена від цілої частини комою.

5.87. Користувач вводить рядок довжиною 20-40 символів. Якщо довжина рядка інша, то програма повинна запитати рядок ще раз. Далі програма повинна вивести на екран буквосполучення з символів 7 і 9, 11 і 26, 30 і 33.

5.88. Користувач вводить слово, яке складається з парної кількості літер. Написати програму, яка міняє місцями першу і другу половини слова.

5.89. Користувач вводить рядок. Написати програму, яка виводить усі символи, що мають парний індекс.

5.90. Користувач вводить рядок, який складається з непарної кількості символів. Програма повинна перевірити довжину рядка на непарність і за потреби запитати рядок знову. Коли введено рядок непарної довжини, програма повинна вивести три символи, які знаходяться в середній частині рядка.

5.91. Написати програму, яка відрізає від рядка, введеного користувачем, перші три і останні три символи.

5.92. Написати програму, яка виводить у стовпчик слово, введене користувачем.

5.93. Користувач вводить слово. Написати програму, яка виводить слово у стовпчик у зворотному порядку.

5.94. Користувач вводить рядок. Програма повинна вивести індекси всіх літер «ю».

5.95. Користувач вводить рядок. Програма повинна вивести у стовпчик всі літери «у», які мають парні індекси.

5.96. З клавіатури вводиться слово. Програма повинна вирізати з нього всі літери «о».

5.97. З клавіатури вводиться рядок. Програма повинна вирізати з нього всі символи «+».

5.98. З клавіатури вводиться рядок. Програма повинна визначити, чи є у рядку символ «е». Якщо такий символ присутній, слід вивести позицію першого включення символу у рядок.

5.99. З клавіатури вводиться рядок. Програма повинна зашифрувати його таким чином: рядок подовжується до тих пір, поки його довжина не стане кратною 5, потім рядок розрізається на слова довжиною 5 символів, слова перевертаються у зворотному порядку і знову склеюються у рядок.

5.100. Користувач вводить рядок. Програма повинна вивести останні 6 символів.

Задачі підвищеної складності

5.101. Користувач вводить рядок. Програма повинна підрахувати найбільшу кількість пробілів, які йдуть у тексті підряд.

5.102. Користувач вводить рядок. Програма повинна підрахувати найбільшу кількість символів, які йдуть у тексті підряд.

5.103. Користувач вводить слово. Програма повинна визначити, скільки різних літер у ньому.

5.104. Користувач вводить слово, у якому є лише дві однакові літери. Програма повинна визначити ці літери.

5.105. Користувач вводить окремо два слова. Програма повинна вивести на екран літери, які зустрічаються в обох словах.

5.106. Користувач вводить окремо два слова. Програма повинна вивести на екран літери, які зустрічаються лише в одному слові.

5.107. Користувач вводить рядок, який містить кілька слів, розділених пробілами. Програма повинна розбити рядок на масив слів.

5.108. Користувач вводить рядок, який складається з кількох слів, розділених пробілами. Програма повинна поміняти місцями перше і останнє слова.

5.109. Користувач вводить рядок, який складається з кількох слів, розділених пробілами. Написати програму, яка виводить на екран слово, яке починається з літери «в».

5.110. Користувач вводить рядок, який складається з кількох слів, розділених пробілами. Програма повинна визначити довжину найкоротшого слова.

5.111. Користувач вводить рядок, який складається з кількох слів, розділених пробілами. Програма повинна визначити довжину найдовшого слова.

5.112. Користувач вводить рядок, який складається з кількох слів, розділених пробілами. Програма повинна вивести окремо слова у порядку спадання їх довжин.

5.113. Користувач вводить рядок, який складається з кількох слів, розділених пробілами. Програма повинна вивести всі слова, які зустрічаються у тексті один раз.

5.114. Написати програму, яка виведе на екран x рядків нулів і одиниць, причому в непарних рядках виводяться нулі, а в парних рядках – одиниці. Кількість символів у рядку задається користувачем.

5.115. Написати програму, яка шифрує повідомлення за таким принципом: після кожного символа вхідного рядка вписуються два символи англійського алфавіту.

5.116. Написати програму, яка дешифрує повідомлення за таким принципом: з рядка вихідного тексту видаляються кожні другий і третій символи, а результат виводиться на екран.

5.117. Написати програму, яка шифрує повідомлення за принципом: перед кожним символом вхідного повідомлення вписується випадковий символ англійського алфавіту.

5.118. Написати програму, яка дешифрує зашифроване повідомлення за таким принципом: залишає кожен другий символ, а символ перед ним видаляє.

5.119. Написати програму, яка шифрує повідомлення за таким принципом: 1) дописує в кінець повідомлення символи «а» до тих пір, поки довжина повідомлення стане кратною 4; 2) кожен групу по 4 символи записує у зворотному порядку і збирає їх в один рядок.

5.120. Написати програму, яка дешифрує повідомлення таким чином: розбиває вихідне повідомлення на групи по 4 символи,

записує кожну групу символів у зворотному порядку і склеює їх в один рядок. Отриманий рядок повинен бути виведений на екран.

5.121. Написати програму, яка шифрує повідомлення за таким принципом: 1) розбиває повідомлення на групи по 2 символи; 2) міняє місцями групи з парними і непарними номерами; 3) склеює групи в один рядок і виводить на екран.

5.122. Написати програму, яка видаляє з рядка символи, які зустрічаються в ньому більше одного разу.

5.123. Користувач вводить два рядки. Програма повинна визначити, скільки разів у рядках зустрічається символа «а».

5.124. Користувач вводить число x – межу діапазону. Програма повинна згенерувати випадкове ціле число в діапазоні $[1;x]$, перетворити його у рядок і повторити цей рядок 5 раз.

5.125. Користувач вводить два рядки символів. Програма повинна скопіювати символи з обох рядків так, щоб символи з першого рядка стали на непарні місця, а символи з другого рядка – на парні місця.

Розділ 6. Одновимірні масиви

Для створення масивів використовується модуль `array`. Масиви подібні до списків, але мають певні обмеження щодо типу даних і розміру кожного елемента.

Масив у мові Python може отримувати такий тип (табл.6.1).

Таблиця 6.1

Типи даних у масивах в Python

Код типу	Тип в C++	Тип в Python	Мінімальний розмір у байтах
'b'	signed char	int	1
'B'	unsigned char	int	1
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2
'I'	unsigned int	int	2
'l'	signed long	int	4
'L'	unsigned long	int	4
'q'	signed long long	int	8
'Q'	unsigned long long	int	8
'f'	float	float	4
'd'	double	float	8

Для створення масиву служить клас `array.array(Туре,[x1,x2,x3])` – так створюється новий масив, у квадратних дужках вказується набір значень. Це може бути список, об'єкт.

Метод `array.typecodes` містить всі типи в масиві.

На відміну від рядків, масиви можна змінювати.

Розглянемо основні методи класу `array` в Python:

Метод `array.typecodes` – символ коду, використаний під час створення масиву.

Метод `array.itemsize` – вказує на розмір у байтах одного елемента у масиві.

Метод `array.append(x)` – дописує в кінець масиву елемент `x`.

Метод `array.buffer_info()` – кортеж (комірка пам'яті, довжина). Використовується під час низькорівневих операцій.

Метод `array.byteswap()` – змінює порядок слідування байтів у кожному елементі масиву.

Метод `array.count(x)` – виводить кількість елементів `x`, які є в масиві.

Метод `array.extend(z)` – додає елемент з об'єкта в масив.

Метод `array.frombytes(b)` – перетворює масив байтів у масив `array`.

Метод `array.fromfile(F,N)` – зчитує `N` елементів з файла і додає їх у кінець масиву. Файл повинен бути відкритим на бінарне читання.

Метод `array.fromlist(list1)` – додає у масив елементи списку `list1`.

Метод `array.index(x)` – виводить номер першого входження елемента `x` у масив.

Метод `array.insert(n,x)` – вставляє на позицію `n` елемент із значенням `x`.

Для роботи з масивами необхідно імпортувати відповідну бібліотеку:

```
from array import *
```

Масив можна оголосити і так:

```
masiv=array('i',[1,2,3,4])
```

Тобто, був оголошений та ініціалізований масив цілих чисел, який складається з 4 елементів. Символ в одинарних лапках означає тип даних (цілий або дійсний), далі вказуються елементи масиву.

Якщо тип даних (`TypeCode`) вказується і (ціле число), то для кожного елемента масиву виділяється 2 байти.

Наприклад:

```
from array import *
masiv = array('i', [1,2,3,4,5])
for i in masiv:
    print(i)
```

У результаті на екран буде виведено:

```
1
2
3
4
5
```

Для оголошення масиву дійсних чисел доцільно використати команду:

```
m=array('f',[2.7,3.9,6.11,2.27,2.59])
```

Доступ до елементів масиву здійснюється за допомогою імені масиву та індексу:

```
from array import *
masiv = array('i', [1,2,3,4,5])
print(masiv[3])
```

Номери елементів масиву починаються з нуля.

Для того, щоб додати до масиву ще один елемент, необхідно використати метод `append()`:

```
masiv=[]  
masiv.append(1)
```

Для того, щоб вставити елемент у масив, використовується метод `insert(index,value)`:

```
massiv.insert(3, 11)
```

У масив вставляється число 11 у місце з індексом 3. При цьому всі елементи масиву, чий індекс дорівнює 3 або більший за нього, зсуваються вправо на один елемент.

Масив можна також розширити, тобто дописати до нього набір значень. Для цього служить метод `extend()`:

```
from array import *  
m1=array('f',[2.7,3.9,6.11,2.27,2.59])  
m2=array('f',[2.2,9.7,6.4,6.3,9.11])  
m1.extend(m2)  
print (m1)
```

На екран буде виведено

```
array('f', [2.700000047683716, 3.90000000953674316,  
6.110000133514404, 2.2699999809265137, 2.5899999141693115,  
2.200000047683716, 9.699999809265137, 6.400000095367432,  
6.300000190734863, 9.109999656677246])
```

Для того, щоб додати елементи зі списку у масив, доцільно скористатися методом `fromlist()`:

```
from array import *  
m2=array('f',[2.2,9.7,6.4,6.3,9.11])  
LL=[2.09,3.6,8.55]  
m2.fromlist(LL)  
print(m2)
```

На екран буде виведено:

```
array('f', [2.200000047683716, 9.699999809265137,  
6.400000095367432, 6.300000190734863, 9.109999656677246,  
2.0899999141693115, 3.5999999046325684, 8.550000190734863])
```

Для видалення елемента масиву служить метод `remove()`:

```
from array import *  
masiv = array('i', [0,1,2,3,4,5])  
masiv.remove(2)  
print(masiv)
```

На екран буде виведено:

```
array('i', [0, 1, 3, 4, 5])
```

Для видалення останнього елемента масиву служить метод `pop()`:

```
from array import *
masiv = array('i', [0,1,2,3,4,5])
masiv.pop()
print(masiv)
```

На екран буде виведено:

```
array('i', [0, 1, 2, 3, 4])
```

Для отримання індекса елемента масиву (якщо відомо його значення) служить метод `index()`:

```
from array import *
masiv = array('i', [17,51,82,33,14,58])
x=masiv.index(51)
print(x)
```

На екран буде виведено:

```
1
```

Цей метод повертає індекс першого входження елемента у масив.

Для зміни порядку елементів у масиві на протилежний служить метод `reverse()`:

```
from array import *
masiv = array('i', [17,51,82,33,14,58])
masiv.reverse()
print(masiv)
```

На екран буде виведено:

```
array('i', [58, 14, 33, 82, 51, 17])
```

Для отримання інформації про буфер масиву слід скористатися методом `buffer_info()`:

```
from array import *
masiv = array('i', [17,51,82,33,14,58])
print(masiv.buffer_info())
```

На екран буде виведено:

```
(39249536, 6)
```

Для того, щоб визначити, скільки раз у масиві зустрічається число, слід використати метод `count()`:

```
from array import *
masiv = array('i', [17,51,82,33,14,58,14])
print(masiv.count(14))
```

На екран буде виведено:

```
2
```

Для перетворення масиву у рядок служить метод `tostring()`:

```
from array import *
masiv = array('i', [17,51,82,33,14,58,14])
s= masiv.tostring()
```

```
print(s)
```

На екран буде виведено:

Для перетворення масиву у список служити метод `tolist()`:

```
from array import *
masiv = array('i', [17,51,82,33,14,58,14])
spisok=masiv.tolist()
print(spisok)
```

На екран буде виведено:

```
[17, 51, 82, 33, 14, 58, 14]
```

Для того, щоб додати у масив символів рядок, слід використати метод `fromstring()`:

```
from numpy import *
from array import *
import numpy as np
masiv3=np.chararray(1)
ryadok="abcdefgh"
masiv3.fromstring(ryadok)
print(masiv3)
```

Розглянемо тепер приклади задач.

Приклади задач

Приклад 6.1. Заповнити масив з 8 елементів такими значеннями: 37,0,50,46,34,46,0,13.

```
import math
a=[37,0,50,46,34,46,0,13]
a[6]=25
for i in range (8):
    print(a[i])
```

Приклад 6.2. Масив складається з 7 випадкових чисел в інтервалі [0;100]. Написати програму, яка виводить у стовпчик всі елементи масиву.

```
import random
masiv=[]
for elem in range (8):
    x=random.randint(0,100)
    masiv.append(x)
for elem in range (8):
    print(masiv[elem])
```

Інший варіант розв'язання

```
import random
array=[0 for i in range(7)]
```

```
for i in range (7):
    array[i]=random.randint(0,100)
    print(array[i])
```

Приклад 6.3. Масив складається з 7 випадкових чисел в інтервалі [0;100]. Програма повинна змінити порядок чисел у масиві на обернений.

```
import random
masiv=[]
x=0
for elem in range (8):
    x=random.randint(0,100)
    masiv.append(x)
for elem in range (8):
    print("1=",masiv[elem])
masiv1=[]
i=7
elem=0
for elem in range (8):
    x=masiv[i]
    masiv1.append(x)
    i=i-1
for elem in range (8):
    print("2=",masiv1[elem])
```

Інший варіант (arra04)

```
from array import *
import random
array=[0 for i in range(7)]
for i in range (7):
    array[i]=random.randint(0,100)
b=array.reverse()
```

Для роботи з масивами можна використати бібліотеку numpy. Її можна викачати з сайту; <https://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>

Завдання для виконання

6.1. Написати програму, яка заповнює масив цілими числами, введеними з клавіатури. Кількість елементів масиву – 8.

6.2. Написати програму, яка заповнює масив випадковими дійсними числами в діапазоні від 0 до 1.

6.3. Написати програму, яка заповнює масив випадковими цілими числами в діапазоні від 10 до 20.

6.4. Заповнити масив з 8 елементів символами «!».

6.5. У масив записані маси учнів (випадкові цілі числа у діапазоні від 30 до 40 кг). Написати програму, яка виводить у стовпчик усі значення, більші за 35.

6.6. Користувач вводить числа a і b – межі діапазону. Програма повинна заповнити масив 10 випадковими цілими числами у цьому діапазоні і вивести їх на екран.

6.7. Написати програму, яка заповнює масив 20 випадковими цілими числами в діапазоні від -20 до 20 і виводить на екран елементи масиву у стовпчик.

6.8. У масив з 20 дійсних чисел занесено дані про напругу різних джерел струму. Написати програму, яка виводить на екран номер і напругу джерела, напруга якого найбільша. Масив ініціалізується на початку програми.

6.9. Користувач вводить число Z . Написати програму, яка наповнить масив з 9 елементів випадковими числами у діапазоні від $-Z$ до Z .

6.10. Масив з 10 цілих чисел містить ціни автомобілів і ініціалізується на початку програми. Написати програму, яка обчислює суму і середнє арифметичне масиву.

6.11. Масив з 6 цілих чисел містить ціни книжок і ініціалізується на початку програми. Написати програму, яка виводить елемент масиву на вибір користувача.

6.12. Масив з 7 дійсних чисел містить ціни олівців і ініціалізується на початку програми. Програма повинна вивести на екран елементи масиву у зворотному порядку.

6.12. Заповнити масив з 16 цілих чисел, які дорівнюють індексу масиву.

6.13. Написати програму, яка заповнює масив спочатку першими 10 натуральними числами, а потім дописує в масив числа 100 і 150.

6.14. Написати програму, яка заповнює масив з 10 елементів натуральними числами від 10 до 1. Використати цикл з передумовою.

6.15. Написати програму, яка заповнює масив цілими числами, які є степенями числа 3 (3^1 , 3^2 , 3^3 тощо). Кількість елементів масиву – 8.

6.16. Написати програму, яка заповнює масив із 10 чисел випадковими числами від 0 до 50 у порядку зростання.

6.17. Написати програму, яка заповнює масив першими 7 членами арифметичної прогресії, для якої $a_1=7$, $d=9$.

6.18. Заповнити масив першими 9 натуральними числами, які діляться одночасно на 8 і на 3.

6.19. Масив містить вартості монет (1 коп, 2 коп, 5 коп тощо) і ініціалізується на початку програми. Написати програму, яка виводить на екран номінали монет вартістю більше 25 коп.

6.20. Масив складається з 12 цілих чисел і ініціалізується на початку програми. Програма повинна вивести на екран спочатку додатні елементи масиву, а потім – від'ємні.

6.21. Масив складається з 9 цілих чисел і містить ціни зошитів. Масив ініціалізується на початку програми. Написати програму, яка виводить другий і четвертий елементи масиву.

6.22. Масив містить 10 випадкових цілих чисел у межах від 0 до 300. Програма повинна вивести всі двозначні числа масиву.

6.23. Масив містить 10 випадкових цілих чисел у межах від 21 до 400. Програма повинна вивести всі тризначні числа масиву.

6.24. Масив з 10 цілих чисел заповнено випадковими числами. Написати програму, яка виводить на екран спочатку непарні елементи, а потім – парні.

6.25. Масив з 16 цілих чисел заповнено випадковими цілими числами в діапазоні від 20 до 200. Написати програму, яка виведе на екран індекси елементів, які закінчуються цифрою 5.

6.26. Масив містить значення опадів у міліметрах за першу декаду квітня і ініціалізується на початку програми. Написати програму, яка виводить на екран номери днів, коли кількість опадів перевищувала середнє арифметичне.

6.27. Масив складається з 12 випадкових дійсних чисел у межах від 0 до 1. Написати програму, яка виведе на екран спочатку елементи з парними індексами, а потім – з непарними.

6.28. У масиві містяться дані про опади за першу декаду серпня. Масив ініціалізується на початку програми. Написати програму, яка виведе на екран дні місяця, в які опадів не було.

6.29. Масив містить дані про масу 11 учнів, які складають футбольну команду, і ініціалізуються на початку програми. Написати програму, яка виводить масу гравця, номер якого задано користувачем.

6.30. Масив містить зріст учнів шкільної волейбольної команди і ініціалізується на початку програми. Написати програму, яка виведе зріст учнів у порядку зростання.

6.31. Масив містить 20 випадкових чисел у діапазоні від X до $-X$. Написати програму, яка заміняє від'ємні елементи масиву їх модулями. Число X вводиться користувачем.

6.32. Масив містить 12 випадкових цілих чисел у діапазоні, заданому користувачем. Написати програму, яка заміняє кожен елемент його квадратом.

6.33. Масив містить 14 дійсних чисел і ініціалізується на початку програми. Написати програму, яка замінює кожен елемент, більший за 20, його квадратним коренем.

6.34. Масив містить 8 дійсних чисел і ініціалізується на початку програми. Написати програму, яка замінить всі елементи з непарними індексами на їх модулі.

6.35. Масив з 7 цілих чисел заповнено випадковими числами в діапазоні від -100 до 100. Програма повинна всі додатні елементи збільшити на q , а всі від'ємні – помножити на z . Числа q і z задає користувач.

6.36. Масив заповнено 11 випадковими цілими числами в діапазоні від -10 до 10. Програма повинна елементи з непарними індексами збільшити на 1, елементи з парними індексами – зменшити на 1.

6.37. Масив заповнено 8 випадковими натуральними числами у межах від 10 до 100. Написати програму, яка подвоює елементи з непарними індексами, а елементи з парними індексами перемножує на 3.

6.38. Масив заповнено 9 випадковими цілими числами в межах від 22 до 88. Написати програму, яка зменшує в 2 рази всі елементи, остання цифра яких дорівнює 4.

6.39. Масив складається з 11 випадкових цілих чисел у діапазоні від 0 до 65. Програма повинна всі парні числа замінити на їх куби, а непарні – перемножити на 12.

6.40. Масив складається з 9 випадкових цілих чисел у діапазоні від 1 до 53. Програма повинна парні числа збільшити на 8, а непарні – зменшити на 11.

6.41. Масив складається з 16 випадкових цілих чисел у діапазоні від 3 до 356. Програма повинна замінити всі числа, кратні 10, на 0.

6.42. Масив складається з 13 випадкових цілих чисел у діапазоні від 12 до 147. Програма повинна всі парні числа поділити на 2, а непарні числа – перемножити на 2.

6.43. Масив складається з 7 випадкових цілих чисел у діапазоні від 63 до 400. Програма повинна непарні елементи зменшити на 10, а парні елементи – перемножити на 10.

6.44. Масив складається з 16 випадкових цілих чисел у діапазоні від 30 до 256. Програма повинна обчислити квадратні корені всіх елементів масиву.

6.45. Масив складається з 15 випадкових цілих чисел у діапазоні від 101 до 406. Програма повинна обчислити середнє арифметичне двох елементів, номери яких задає користувач.

6.46. Масив складається з 7 випадкових цілих чисел у діапазоні від 1 до 33. Програма повинна обчислити добуток всіх елементів масиву.

6.47. Масив складається з 9 випадкових цілих чисел у діапазоні від 23 до 58. Програма повинна обчислити суму квадратів усіх елементів масиву.

6.48. Масив складається з 11 випадкових цілих чисел у діапазоні від 16 до 406. Програма повинна обчислити суму перших 4 елементів масиву.

6.49. Масив складається з 22 випадкових цілих чисел у діапазоні від 1 до 155. Програма повинна обчислити суму елементів масиву, які мають індекси від a до b . Числа a і b вводяться користувачем ($a < b$).

6.50. Масив складається з 8 випадкових цілих чисел у діапазоні від 11 до 55. Програма повинна обчислити середнє арифметичне всіх елементів масиву.

6.51. Масив складається з 16 випадкових цілих чисел у діапазоні від 1 до 55. Програма повинна обчислити середнє арифметичне елементів масиву, які мають індекси від $A1$ до $A2$ ($A1 < A2$). Числа $A1$ і $A2$ задаються користувачем.

6.52. У масиві містяться дані про виробництво деталей за перший тиждень квітня. Масив ініціалізується на початку програми. Написати програму, яка обчислює загальну кількість виготовлених деталей.

6.53. У масиві зберігаються дані про випуск стільців на меблевій фабриці. Масив ініціалізується на початку програми. Програма повинна визначити кількість елементів, більших за середнє арифметичне.

6.54. У масив записано опори 16 резисторів, з'єднаних послідовно (випадкові цілі числа у діапазоні від 20 до 160). Написати програму, яка обчислить загальний опір резисторів.

6.55. Масив містить дані про місткість автобусів і складається з 10 випадкових чисел у діапазоні від 12 до 60. Написати програму, яка підраховує кількість автобусів, які вміщають менше 30 пасажирів.

6.56. Масив з 30 чисел містить дані про опади, які випали в червні (випадкові числа в межах від 0 до 200). Написати програму, яка обчислить кількість опадів за кожну декаду червня.

6.57. Масив з 31 цілого числа містить дані про опади у липні (випадкові числа в діапазоні від 0 до 80). Написати програму, яка обчислить середню кількість опадів за день і номер дня, за який випало найбільше опадів. Якщо таких днів кілька, то вивести номери всіх днів.

6.58. Масив складається з 10 випадкових цілих чисел у діапазоні від -20 до 20. Написати програму, яка визначить, чи є сума всіх елементів масиву від'ємним числом.

6.59. Масив містить дані про кількість книг у кожному з 18 відділів бібліотеки (випадкові числа в діапазоні від 80 до 450). Написати програму, яка визначить, чи є кількість книг у всій бібліотеці чотиризначним числом.

6.60. Вантажопідйомність автомобіля створює Z кг (задається користувачем). У кузов автомобіля завантажуються 12 предметів, масу кожного з яких користувач заносить у масив. Програма повинна визначити, чи вистачить вантажопідйомності автомобіля для перевезення всього вантажу.

6.61. Клас складається з 12 учнів, які отримали оцінки за нормативи з фізкультури (випадкові числа в діапазоні від 4 до 12 балів). Написати програму, яка порівняє середні арифметичні перших шести учнів і других шести учнів і виведе результат.

6.62. Масив містить дані про вартість меблів і складається з 7 випадкових цілих чисел в діапазоні від 650 до 3500. Написати програму, яка обчислить загальну вартість меблів і визначить вартість найдешевшого предмета з меблів.

6.63. Масив містить 9 значень – маси велосипедів (вводяться користувачем). Програма повинна вивести маси 3 найважчих велосипедів.

6.64. Масив з 15 чисел містить значення температури в градусах Цельсія за першу половину січня (вводяться користувачем). Написати програму, яка підраховує кількість днів, коли температура була більшою від нуля.

6.65. Масив з 17 чисел містить випадкові числа в діапазоні від -10 до 20. Написати програму, яка обчислює частку від ділення суми додатніх елементів масиву на суму від'ємних.

6.66. Масив містить 25 чисел, які являють собою втрати тепла у мегаджоулях у різних будинках (випадкові числа в діапазоні від 0 до 45). Програма повинна визначити, скільки будинків мають втрати тепла більші, ніж 30 мегаджоулів.

6.67. Масив містить дані про чисельність мешканців у 32 будинках на одній з вулиць міста (випадкові числа у діапазоні від 100 до 600), причому одна сторона вулиці містить будинки з непарними номерами, а інша – з парними номерами. Написати програму, яка визначить, на якій стороні вулиці живе більше мешканців.

6.68. Масив містить дані про температуру повітря за першу половину березня і складається з випадкових чисел у діапазоні від -7 до +8. Програма повинна визначити кількість днів, коли не було морозів.

6.69. Масив містить оцінки 20 учнів з геометрії (випадкові числа в діапазоні від 1 до 12). Програма повинна визначити кількість учнів, які мають низькі оцінки (менше за 4).

6.70. Дані про продаж банок фарби однієї з торгових фірм за травень місяць занесено у масив (випадкові числа в діапазоні від 2 до 50). Програма повинна вивести на екран дані за ті дні, коли кількість проданих банок фарби перевищила Z . Число Z вводиться користувачем.

6.71. Вартість 12 мобільних телефонів занесено у масив (випадкові числа в діапазоні від 40 до 6000). Написати програму, яка виведе ціни, які перевищують K . Число K вводиться користувачем.

6.72. Масив містить дані про зріст 18 учнів (випадкові числа в діапазоні від 120 до 160 см). Програма повинна вивести зріст учнів, який перевищує 140 см. Якщо таких учнів немає, програма повинна вивести повідомлення.

6.73. Масив містить дані про тривалість життя ізотопів водню у мілісекундах і складається з 12 цілих чисел. Користувач задає діапазон номерів елементів, програма повинна вивести значення цих елементів на екран.

6.74. Масив містить результати 7 ігор волейбольної команди (випадкові числа в діапазоні від 0 до 3). Програма повинна вивести на екран: 1) суму балів, набрану командою; 2) кількість програшів (нульові значення масиву); 3) кількість вигравшів (значення елементів масиву «3»).

6.75. Масив містить оцінки одного з учнів з 9 предметів (випадкові числа в діапазоні від 4 до 12). Програма повинна визначити, скільки оцінок вказують на високий рівень (10-12 балів).

6.76. Масив містить значення температури повітря у грудні (випадкові числа в діапазоні від -20 до +3). Програма повинна визначити: 1) кількість днів, коли температура перевищувала 0 градусів; 2) кількість днів з від'ємною температурою.

6.77. Масив містить результати збирання пшениці в тоннах за день у липні (14 випадкових чисел в діапазоні від 0 до 40). Програма повинна: 1) визначити день, в який було досягнуто найвищих показників; 2) визначити день, в який продуктивність роботи дорівнювала нулю.

6.78. Масив містить результати випробувань на міцність заготовок на заводі і складається з 10 випадкових чисел у діапазоні від 12 до 25. Написати програму, яка визначить: 1) скільки чисел закінчуються на цифру 5; 2) скільки чисел закінчуються на цифру 7.

6.79. Оцінки учнів за контрольну роботу записані в масив (18 чисел у діапазоні від 4 до 12, масив ініціалізується на початку програми). Програма повинна визначити: 1) кількість учнів, які

показали високий рівень (10-12 балів); 2) кількість учнів, які показали достатній рівень (7-9 балів).

6.80. Массив містить дані про ремонт дисків коліс у автомайстерні за кожен день (30 випадкових чисел у діапазоні від 0 до 22). Програма повинна визначити дні, коли майстерня два дні підряд ремонтувала парну кількість дисків.

6.81. Користувач вводить массив з 10 цілих чисел. Програма повинна підрахувати кількість пар чисел, що закінчуються на 0.

6.82. Користувач вводить массив з 12 чисел. Програма повинна визначити кількість елементів масиву, більших за попередні.

6.83. Дано массив з 7 дійсних чисел. Програма повинна підрахувати кількість чисел, більших за 27,09.

6.84. Користувач вводить массив з 9 дійсних чисел. Написати програму, яка визначає медіану масиву.

6.85. Користувач вводить массив з 5 цілих чисел. Програма повинна підрахувати кількість елементів, більших від 0.

6.86. Массив з 25 цілих чисел містить зріст учнів класу у сантиметрах. Програма повинна підрахувати кількість учнів, вищих за 170 см.

6.87. Массив з 31 числа містить дані про виробництво борошна на хлібокомбінаті. Написати програму, яка обчислить виробництво борошна за весь місяць.

6.88. Массив з 30 чисел містить дані про виробництво борошна на хлібокомбінаті. Написати програму, яка визначить кількість днів, коли виробництво борошна перевищило 2 т.

6.89. Массив містить дані про температуру за листопад. Програма повинна обчислити окремо середнє арифметичне додатних і середнє арифметичне від'ємних чисел.

6.90. Массив з 11 дійсних чисел містить маси людей. Написати програму, яка обчислить кількість людей, чия маса перевищує 90 кг.

6.91. Массив з 8 чисел містить вік співробітників відділу у роках (випадкові числа у межах від 30 до 55). Написати програму, яка обчислює середній вік співробітників.

6.92. Массив з 12 чисел містить вік учителів у роках (випадкові числа у межах від 24 до 60). Написати програму, яка обчислює різницю у віці між найстаршим і наймолодшим учителями.

6.93. Массив з 6 чисел містить швидкості літаків (випадкові числа у діапазоні [600; 950]). Написати програму, яка визначить, у скільки разів швидкість найшвидшого літака перевищує швидкість найповільнішого.

6.94. Массив з 9 цілих чисел містить оцінки учнів з шкільних предметів. Написати програму, яка обчислює середній бал учня.

6.95. Масив з 7 дійсних чисел містить дані про напругу на клеммах акумулятора в різні моменти часу. Написати програму, яка визначить середню напругу на клеммах акумулятора.

6.96. Масив з 5 дійсних чисел містить дані про урожайність різних культур з 1 га. Написати програму, яка обчислить найбільший урожай з поля. Площу поля вводить користувач.

6.97. Два масиви, кожен з 7 дійсних чисел, містять відповідно дані про площу полів (га) і урожайність (ц/га) кожного поля. Написати програму, яка обчислить загальний урожай, зібраний з цих полів.

6.98. Масив містить дані про атмосферний тиск у різні дні і містить 14 цілих чисел (випадкові числа у межах від 735 до 770), які відповідають тиску за першу половину лютого. Написати програму, яка виведе числа місяця, в які тиск був найбільшим.

6.99. У масив записано атмосферний тиск за перший тиждень квітня (випадкові числа у межах від 740 до 760). Написати програму, яка буде визначати середній тиск за цей період.

6.100. Масив містить дані про опади за останню декаду жовтня (випадкові числа у межах від 0 до 50). Написати програму, яка обчислить середню кількість опадів.

Задачі підвищеної складності

6.101. У таблицю записано кількість опадів за другу декаду вересня. Написати програму, яка відсортуює масив за зростанням.

6.102. Масив містить 12 випадкових чисел у діапазоні від 1 до 20. Програма повинна визначити, скільки разів зустрічаються два однакові числа, які йдуть підряд.

6.103. Масив складається з 11 випадкових цілих чисел. Програма повинна обчислити різницю між сусідніми елементами масиву.

6.104. Таблиця містить дані про випуск мила одним із заводів за тиждень. Програма повинна упорядкувати таблицю за спаданням.

6.105. Таблиця містить дані про роки народження співробітників фірми (числа від 1960 до 2000, масив ініціалізується на початку програми). Програма повинна вирахувати вік співробітників і відсортувати дані у порядку спадання.

6.106. Таблиця містить дані про рік випуску 11 автомобілів (випадкові числа від 1995 до 2021). Написати програму, яка випише в другу таблицю роки випуску автомобілів, які мають вік понад 8 років.

6.107. Масив містить дані про рік випуску мотоциклів різних марок. Програма повинна упорядкувати масив за принципом «від меншого до більшого» і вивести масив на екран.

6.108. Масив містить дані про рік виготовлення мотоциклів (випадкові числа у діапазоні від 2000 до 2020). Написати програму, яка упорядкує за віком мотоцикли.

6.109. У дві таблиці записано дані: відповідно довжини і висоти прямокутників (випадкові числа у діапазоні від 1 до 8). Написати програму, яка виведе на екран найбільшу площу прямокутника.

6.110. Два масиви містять дані про опади відповідно в першу і другу половину лютого. Програма повинна записати ці дані в один масив і вивести на екран найбільше і найменше значення.

6.111. Користувач вводить 7 чисел масиву (кількість проданих автомобілів різними фірмами). Програма повинна визначити, чи існують дві фірми, продаж автомобілів яких однаковий.

6.112. Дані про опади у червні занесено у три одновимірні масиви (випадкові числа у діапазоні [0;200]). Програма повинна об'єднати масиви в один і вивести число місяця, коли рівень опадів найменший.

6.113. Дані про прибутки фірми за червень (випадкові числа у діапазоні від 10000 до 25000) занесено в одновимірний масив. Програма повинна розбити цей масив на три однакові і вивести на екран найбільші значення опадів за першу, другу і третю декади червня.

6.114. Кількість деталей, виготовлених у одному з цехів заводу за вересень, занесено у масив (випадкові числа у діапазоні від 20 до 50). Програма повинна виписати кількість деталей, виготовлених по непарних числах, в один масив, а кількість деталей, виготовлених по парних числах, у другий масив, і вивести обидва масиви кожен в рядок на екран.

6.115. Таблиця містить дані про кількість викликів доставки піци з 12 будинків на вулиці. Написати програму, яка видалить з таблиці усі нульові значення.

6.116. У таблицю записано кількість доставок з інтернет-магазину за першу половину березня. Програма повинна розбити таблицю на 3 рівні частини і визначити, в якій з цих частин кількість викликів більша.

6.117. Таблиця містить дані про частоту, з якою зустрічаються різні літери англійського алфавіту у тексті оповідання (26 випадкових чисел у діапазоні від 100 до 1000). Програма повинна визначити повний обсяг тексту.

6.118. Таблиця містить дані про частоту, з якою зустрічаються різні літери українського алфавіту у тексті оповідання (32 випадкових числа у діапазоні від 30 до 3000). Програма повинна визначити номери трьох літер, які зустрічаються найчастіше.

6.119. Перша таблиця містить густини металів. Друга таблиця містить об'єми деталей, виготовлених з цих деталей. Програма повинна вивести найбільшу і найменшу маси тіл.

6.120. Одна з систем шифрування даних працює за таким алгоритмом: масив вихідних чисел ділиться на два послідовних масива, після чого перетворюються в один масив, вставляючи в нього числа по чергово з першого та другого масивів. Реалізувати даний алгоритм для масиву довжиною 12 елементів.

6.121. Одна з систем шифрування даних працює таким чином: після кожного числа вихідного повідомлення вставляються випадкові числа у діапазоні від 1 до 50. Реалізувати даний алгоритм для повідомлення, що складається з 12 чисел.

6.122. Одна з систем шифрування функціонує за таким принципом: після кожного числа x початкового повідомлення дописується випадкове число у діапазоні від 1 до x . Реалізувати цей алгоритм для масиву чисел довжиною 16 чисел. Початкове повідомлення у вигляді рядка чисел, розділених пробілами, зчитується з клавіатури.

6.123. Масив $A[]$ заповнено 9 степенями числа 2. Масив $B[]$ заповнено випадковими цілими числами у діапазоні від 1 до 256. Написати програму, яка запише добутки елементів цих масивів у масив $C[]$.

6.124. Написати програму, яка розбиває початковий масив на два однакових масиви, сортує перший з них за зростанням, а другий за спаданням і знову об'єднує обидва масиви в один.

6.125. Написати програму, яка розбиває початковий масив на чотири однакових масивів, сортує їх за зростанням і об'єднує в один.

Розділ 7. Двовимірні масиви

Одновимірний і двовимірний можна створювати і обробляти за допомогою бібліотеки NumPy. Бібліотека містить широкий набір пакетів для розв'язування різних завдань. З її допомогою можна створювати не лише масиви, а й однорідні матриці. Для того, щоб підключити бібліотеку, слід імпортувати її:

```
import numpy as np
```

Для створення масиву можна використати метод `array()`:

```
import numpy as np
masiv=np.array([0,2])
print(masiv[1])
```

На екран буде виведено:

```
2
```

Для перевірки типу масиву застосовується метод `array.type()`:

```
masiv=np.array([0,2])
print(masiv[1])
print (array.type(masiv))
```

Тип масиву можна встановити під час створення:

```
masiv=np.array([0,2],dtype=np.int)
```

Можна заповнити масив нулями:

```
masiv2=array('i',[ ])
np.zeros(masiv2)
```

Для створення двовимірного масиву можна використати команди:

```
import numpy as np
masiv = np.array([[1, 2, 3], [4, 5, 6]])
print(masiv)
print(masiv[1][2])
```

На екран буде виведено:

```
[[1 2 3]
 [4 5 6]]
3
```

Розглянемо операції з масивами і матрицями, які можна виконати за допомогою бібліотеки NumPy.

Додавання матриць. Матриці повинні мати однаковий розмір.

```
import numpy as np
masiv1 = np.array([[1, 2, 3], [4, 5, 6]])
masiv2 = np.array([[9, 12, 13], [14, 15, 16]])
masiv=np.array([[0,0,0],[0,0,0]])
masiv=masiv1+masiv2
```

```
print(masiv)
```

На екран буде виведено:

```
[[10 14 16]
 [18 20 22]]
```

Для заповнення масиву значеннями можна використати цикл:

```
import numpy as np
masiv=np.array([[0,0,0],[0,0,0]])
print(masiv)
for i in range(2):
    for j in range(3):
        masiv[i][j]=7;
print(masiv)
```

На екран буде виведено:

```
[[0 0 0]
 [0 0 0]]

[[7 7 7]
 [7 7 7]]
```

Для знаходження суми двох двовимірних масивів або матриць можна використати вбудовану функцію:

```
from numpy import *
import numpy as np
masiv1=np.array([[1,2,3],[4,5,6]])
masiv2=np.array([[11,12,13],[14,15,16]])
masiv3=np.array(masiv1+masiv2)
print(masiv3)
```

На екран буде виведено:

```
[[12 14 16]
 [18 20 22]]
```

Для того, щоб визначити розмірність масиву, служить метод shape():

```
from numpy import *
import numpy as np
masiv1=np.array([[1,2,3],[4,5,6]])
masiv2=np.array([[11,12,13],[14,15,16]])
masiv3=np.array(masiv1+masiv2)
print(np.shape(masiv3))
```

На екран буде виведено:

```
(2, 3)
```

Розглянемо властивості масиву в NumPy.

Властивість `ndarray.ndim` – розмірність масиву (одно- або двовимірний). Приклад:

```
from numpy import *
import numpy as np
masiv1=np.array([[1,2,3],[4,5,6]])
print(masiv1.ndim)
```

Виведений результат – 2 (тобто, масив двовимірний).

Метод `shape()` можна використати і як властивість. Для цього слід вказати `shape` без дужок. Аналогічно – властивість `size`:

```
from numpy import *
import numpy as np
masiv1=np.array([[1,2,3],[4,5,6]])
masiv2=np.array([[11,12,13],[14,15,16]])
masiv3=np.array(masiv1+masiv2)
print(masiv1.shape)
print(masiv1.size)
```

На екран буде виведено:

```
(2, 3)
6
```

Обсяг масиву (`size`) – це загальна кількість елементів масиву (у даному випадку $6=2 \times 3$).

Властивість `dtype` описує тип елементів масиву:

```
from numpy import *
import numpy as np
masiv1=np.array([[1,2,3],[4,5,6]])
print(masiv1.dtype)
```

Програма виведе

```
int32
```

Властивість `dtype` може набувати таких значень: `np.int`, `np.float`, `np.str` (рядковий).

Властивість `itemsize` повертає розмір кожного елемента масиву у байтах. Наприклад:

```
from numpy import *
import numpy as np
masiv1=np.array([[1,2,3],[4,5,6]])
print(masiv1.itemsize)
```

Виведене значення – 4 ($=32:8$).

Властивість `ndarray.data` повертає буфер з актуальними елементами масиву. Так можна отримати доступ до елементів масиву через індекси.

```
import numpy as np
masiv1=np.array([[1,2,3],[4,5,6]])
```

```
print(masiv1.data)
```

Програма виведе:

```
<memory at 0x02F2E828>
```

Метод `ndarray.sum()` – виводить суму всіх елементів масиву:

```
from numpy import *  
import numpy as np  
masiv1=np.array([[1,2,3],[4,5,6]])  
print(masiv1.sum())
```

Виведене значення – 21.

Метод `ndarray.min()` – повертає значення мінімального елемента з масиву. Наприклад:

```
from numpy import *  
import numpy as np  
masiv1=np.array([[1,2,3],[4,5,6]])  
print(masiv1.min())
```

Виведене значення – 1.

Метод `ndarray.max()` повертає значення максимального елемента масиву:

```
from numpy import *  
import numpy as np  
masiv1=np.array([[1,2,3],[4,5,6]])  
print(masiv1.max())
```

Виведене значення – 6.

Розглянемо функції бібліотеки NumPy

Функція `type (numpy.ndarray)` повертає тип параметра:

```
from numpy import *  
import numpy as np  
masiv1=np.array([[1,2,3],[4,5,6]])  
print(type(masiv1))
```

На екран виводиться:

```
<class 'numpy.ndarray'>
```

Функція `numpy.zeros()`, `numpy.zeros((rows,columns), dtype)` створює масив з заданою кількістю вимірів, кожний елемент якого дорівнює 0. Наприклад:

```
import numpy as np  
masiv=np.zeros((3,3))  
print(masiv)
```

Програма виводить:

```
[[0. 0. 0.]  
 [0. 0. 0.]  
 [0. 0. 0.]]
```

Метод `numpy.ones()`, `numpy.ones((rows, columns), dtype)` створює масив з заданою кількістю вимірів, кожен елемент якого буде дорівнювати 1. Якщо тип `dtype` не вказано, то за замовчуванням буде використовуватися `dtype`.

Наприклад:

```
from numpy import *
import numpy as np
masiv=np.ones((3,3))
print(masiv)
```

Програма виведе на екран:

```
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
```

Функція `numpy.empty()`, `numpy.empty((rows, columns))` створює масив, кожен елемент якого буде випадковим:

```
from numpy import *
import numpy as np
masiv=np.empty((3,3))
print(masiv)
```

Програма виведе на екран:

```
[[3.89941022e-288 3.69745581e-288 2.97583931e-288]
 [2.46442878e-288 1.64350890e-288 2.97748805e-288]
 [4.10463706e-288 2.29815644e-311 1.39629327e+084]]
```

Функція `numpy.arange()`, `numpy.arange(start, stop, step)` застосовується для створення масиву `numpy`, значення елементів якого лежать у межах від `start` до `stop` з різницею, що дорівнює `step`:

```
from numpy import *
import numpy as np
masiv=np.arange(3,33,5)
print(masiv)
```

Програма виводить на екран:

```
[ 3  8 13 18 23 28]
```

Функція `numpy.linspace()`, `numpy.linspace(start, stop, number)` створює масив `numpy`, значення елементів якого знаходяться в діапазоні значень від `start` до `stop`, `number` – це кількість елементів масиву. Тип за замовчуванням – `float64`. Наприклад:

```
from numpy import *
import numpy as np
masiv=np.linspace(3,33,5)
print(masiv)
```

Програма виводить на екран:

[3. 10.5 18. 25.5 33.]

Функція `numpy.logspace()`, `numpy.logspace(start, stop, number)` створює масив `numpy`, значення елементів якого знаходяться в межах від `start` до `stop`, а розмір масиву встановлюється параметром `number`. Тип за замовчуванням – `float64`. Всі елементи знаходяться в межах логарифмічної шкали (тобто є логарифмами відповідних елементів).
Наприклад:

```
from numpy import *
import numpy as np
masiv=np.logspace(3,33,5)
print(masiv)
```

Програма виводить:

```
[1.00000000e+03 3.16227766e+10 1.00000000e+18 3.16227766e+25
1.00000000e+33]
```

Функція `numpy.sin()` обчислює синус, функція `numpy.cos()` – косинус, функція `numpy.tan()` – тангенс. Наприклад:

```
from numpy import *
import numpy as np
masiv=np.logspace(3,33,5)
print(np.sin(masiv))
```

Програма виводить:

```
[ 0.82687954  0.31675523 -0.9928161 -0.80909158  0.70104291]
```

Метод `numpy.reshape()`, `numpy.reshape(dims)` застосовується для зміни розрядності масиву (одно- або двовимірний). Наприклад:

```
from numpy import *
import numpy as np
masiv=np.logspace(3,33,5)
print(np.reshape(3,3))
```

Програма виводить:

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

Примітка. Для того, щоб змінити розмірність масиву, масив до перетворення і після перетворення повинен мати однакову кількість елементів.

Метод `numpy.random.random()`, `numpy.random.random((rows, cols))` повертає масив з заданою кількістю вимірів, у якому кожен елемент є випадковим числом. Наприклад:

```
from numpy import *
```

```
import numpy as np
masiv=np.random.random((3,3))
print(masiv)
```

Програма виводить:

```
[[0.03570564 0.24613438 0.352443 ]
 [0.2371729  0.75811104 0.15044101]
 [0.28391057 0.4235592  0.29394144]]
```

Функція `numpy.exp()`, `numpy.exp(masiv)` обчислює експоненти всіх елементів масиву. Наприклад:

```
from numpy import *
import numpy as np
masiv=np.linspace(3,30,4)
e=np.array([[0,0,0],[0,0,0]])
e=np.exp(masiv)
print(e)
```

Програма виводить:

```
[2.00855369e+01 1.62754791e+05 1.31881573e+09 1.06864746e+13]
```

Функція `numpy.sqrt()`, `numpy.sqrt(masiv)` повертає масив з квадратним коренем кожного елемента. Наприклад:

```
from numpy import *
import numpy as np
masiv=np.linspace(3,30,4)
sq=np.sqrt(masiv)
print(sq)
```

Програма виводить:

```
[1.73205081 3.46410162 4.58257569 5.47722558]
```

Базові операції бібліотеки NumPy

Додавання двох масивів (матриць):

```
from numpy import *
import numpy as np
masiv1=np.array([[1,1,1],[2,2,2]])
masiv2=np.array([[3,3,3],[4,4,4]])
masiv3=masiv1+masiv2
print(masiv3)
```

Програма виводить:

```
[[4 4 4]
 [6 6 6]]
```

Різниця двох масивів (матриць):

```
from numpy import *
```

```
import numpy as np
masiv1=np.array([[1,1,1],[2,2,2]])
masiv2=np.array([[3,3,3],[4,4,4]])
masiv3=masiv1-masiv2
print(masiv3)
```

Програма виводить:

```
[[ -2 -2 -2]
 [ -2 -2 -2]]
```

Добуток двох масивів (матриць):

```
from numpy import *
import numpy as np
masiv1=np.array([[1,1,1],[2,2,2]])
masiv2=np.array([[3,3,3],[4,4,4]])
masiv3=masiv1*masiv2
print(masiv3)
```

Програма виводить:

```
[[3 3 3]
 [8 8 8]]
```

Матричний добуток двох масивів обчислюється одним із двох способів:

- 1) `masiv3=masiv1@masiv2`
- 2) `masiv1.dot(masiv2)`

Матриці повинні бути квадратними (2x2, 3x3 або 4x4).

Приклад програми:

```
from numpy import *
import numpy as np
masiv1=np.array([[1,1,1],[2,2,2],[3,3,3]])
masiv2=np.array([[3,3,3],[4,4,4],[5,5,5]])
masiv3=masiv1@masiv2
print(masiv3)
print('*****')
print(masiv1.dot(masiv2))
```

Програма виводить:

```
[[12 12 12]
 [24 24 24]
 [36 36 36]]
*****
[[12 12 12]
 [24 24 24]
 [36 36 36]]
```

Приклади задач

Приклад 7.1. Написати програму, яка заповнює масив 2x3 випадковими числами.

Розв'язання

```
import numpy as np
import random
masiv=np.array([[0,0,0],[0,0,0]])
for i in range(2):
    for j in range(3):
        masiv[i][j]=random.randint(0,9)
print(masiv)
```

Приклад 7.2. Написати програму, яка заповнює масив 3x3 випадковими числами і виводить його на екран по рядках.

Розв'язання

```
import numpy as np
import random
masiv=np.array([[0,0,0],[0,0,0],[0,0,0]])
for i in range(3):
    for j in range(3):
        masiv[i][j]=random.randint(0,20)
for i in range(3):
    print(masiv[i])
```

Приклад 7.3. Написати програму, яка виводить на екран суми рядків масиву 4x3, наповненого випадковими числами.

Розв'язання

```
import numpy as np
import random
masiv=np.array([[0,0,0],[0,0,0],[0,0,0],[0,0,0]])
for i in range(4):
    for j in range(3):
        masiv[i][j]=random.randint(0,11)
print(masiv)
for i in range(4):
    print(masiv[i].sum())
```

Приклад 7.4. У таблицю 2x2 занесено випадкові числа у діапазоні [1;9]. Написати програму, яка розширює масив до розмірів 2x7 і дописує в першому рядку випадкові числа у діапазоні від 10 до 20, у другому рядку – випадкові числа у діапазоні від 21 до 30.

Розв'язання

```
import numpy as np
import random
masiv=np.ones ((2,2),dtype=np.int)
for i in range (2):
    for j in range (2):
```

```
masiv[i][j]=random.randint (1,9)
print (masiv)
masiv=masiv.reshape(4)
print (masiv)
masiv2=np.ones ((14),dtype=np.int)
for i in range (4):
    masiv2[i]=masiv[i]
print (masiv2)
for i in range (4,7,1):
    masiv2[i]=random.randint (10,20)
for i in range (7,14,1):
    masiv2[i]=random.randint (21,30)
print (masiv2)
masiv2=masiv2.reshape(2,7)
print (masiv2)
```

Задачі

- 7.1. Написати програму, яка наповнює масив 3x3 нулями.
- 7.2. Написати програму, яка наповнює масив 2x6 випадковими числами.
- 7.3. Написати програму, яка наповнює масив 3x4 одиницями.
- 7.4. Створити масив 2x3 і заповнити його числами від 1 до 6.
- 7.5. Створити масив 3x3 і заповнити його під час створення довільними дійсними числами.
- 7.6. Створити масив 2x2 і заповнити його під час створення довільними дійсними числами.
- 7.7. Створити масив 3x4 і заповнити його під час створення довільними цілими числами.
- 7.8. Створити масив 2x5 і заповнити його під час створення довільними цілими числами.
- 7.9. Створити масив 3x3 і перетворити його на масив 1x9.
- 7.10. Написати програму для додавання двох масивів 2x2.
- 7.11. Написати програму для додавання двох масивів 3x3.
- 7.12. Написати програму для додавання двох масивів 2x4.
- 7.12. Написати програму для додавання двох масивів 3x4.
- 7.12. Створити масив 3x3 і визначити його найменший елемент.
- 7.13. Створити масив 2x2 і визначити його найменший елемент.
- 7.14. Створити масив 3x3 і заповнити його числами 22 під час створення.
- 7.15. Створити масив 2x3 і обчислити суму його елементів.

7.16. Створити масив 3×3 і обчислити його найбільший елемент.

7.17. Створити масив 4×4 і визначити його найменший елемент.

7.18. Створити масив 4×4 і заповнити його випадковими цілими числами. Написати програму, яка виводить значення елементів масиву, які знаходяться у кутах таблиці.

7.19. Створити масив 3×3 і заповнити його випадковими числами. Написати програму, яка обчислює суму елементів, які знаходяться в середньому рядку масиву.

7.20. Створити масив 4×4 і заповнити його випадковими числами. Написати програму, яка обчислить суму елементів масиву, які знаходяться в четвертому стовпчику.

7.21. Створити масив 2×6 і заповнити його довільними цілими числами під час створення. Написати програму, яка обчислює суму елементів масиву.

7.22. Створити масив 2×4 і заповнити його випадковими числами. Написати програму, яка обчислить суму елементів масиву.

7.23. Створити масив 3×3 і знайти суму елементів, які лежать на головній діагоналі.

7.24. Створити масив 2×7 , заповнити його випадковими цілими числами і вивести на екран третій стовпчик масиву.

7.25. Створити масив 2×4 , заповнити його довільними цілими числами під час створення. Написати програму, яка обчислить суму елементів масиву, менших від 0.

7.26. Створити масив 2×3 . Заповнити його числами, введеними з клавіатури. Написати програму, яка обчислить суму елементів правого стовпчика.

7.27. Створити масив 3×2 і заповнити його довільними дійсними числами під час створення. Написати програму, яка виведе на екран найбільший елемент другого рядка.

7.28. Створити масив 2×2 і заповнити його випадковими цілими числами у межах від 0 до 22. Написати програму, яка визначить найбільший елемент масиву.

7.29. Створити масив 4×4 і заповнити його випадковими цілими числами у межах від 0 до 20. Написати програму, яка обчислить суму елементів останнього рядка масиву.

7.30. Створити масив 5×5 і заповнити його випадковими цілими числами у межах від 0 до 40. Написати програму, яка виведе найменший елемент середнього стовпчика.

7.31. Створити масив 3×6 і заповнити його нулями. Написати програму, яка виводить загальну кількість елементів масиву і його розмірність.

7.32. Написати програму, яка визначає найменший елемент другого рядка масиву 4×4 , заповненого випадковими цілими числами в діапазоні від 0 до 50.

7.33. Написати програму, яка обчислить суму елементів середнього рядка масиву 5×3 , заповненого випадковими числами в межах від 0 до 60.

7.34. Написати програму, яка заповнює масив 3×3 випадковими числами і знаходить суму елементів, що знаходяться на головній діагоналі.

7.35. Написати програму, яка заповнює масив 3×3 випадковими числами і знаходить найбільший елемент, що знаходяться на допоміжній діагоналі.

7.36. Масив 4×4 заповнено випадковими числами. Написати програму, яка обчислить суму елементів масиву, які знаходяться в кутах таблиці.

7.37. Масив 3×5 заповнено випадковими числами. Програма повинна поміняти місцями перший і останній рядки масиву.

7.38. Масив 2×4 заповнено випадковими цілими числами в діапазоні від 1 до 10. Програма повинна поміняти місцями перший і останній стовпчики масиву.

7.39. Масив 5×4 містить стовпчики значень довжин сторін прямокутників (a_1, b_1, a_2, b_2). Програма повинна вивести у два стовпчики площі прямокутників.

7.40. Масив 7×4 містить стовпчики значень довжин сторін прямокутників (a_1, b_1, a_2, b_2). Написати програму, яка виведе найбільшу площу прямокутника з першого масиву і найбільшу площу з другого масиву.

7.41. Масив 3×3 містить випадкові числа у діапазоні $[0;20]$. Програма повинна обчислити кількість нульових значень у масиві.

7.42. Масив 3×6 заповнений випадковими числами. Написати програму, яка виведе на екран індекс рядка, сума елементів якого найбільша.

7.43. Масив 5×3 містить довжини ребер прямокутних паралелепіпедів. Програма повинна вивести номер паралелепіпеда, об'єм якого найбільший.

7.44. Масив 7×2 містить розміри a і b полів, які мають форму прямокутника (задано на початку програми). Користувач вводить урожайність культури, яка вирощується на цих полях. Програма повинна вивести номер поля, урожай з якого найбільший.

7.45. Масив 4×2 містить лінійні розміри кімнат (по стовпчиках – довжину і ширину), які мають форму прямокутника. На фарбування одного квадратного метра підлоги витрачається 180 г фарби.

Програма повинна обчислити масу фарби, яка потрібна для фарбування всіх кімнат.

7.46. Масив 6×2 містить лінійні розміри стін приміщень, які мають форму прямокутника. На фарбування одного шару стіни площею 1 м^2 потрібно 210 г фарби. Написати програму, яка обчислить загальну масу фарби в кілограмах.

7.47. Масив 9×2 містить розміри прямокутників (випадкові числа у межах від 2 до 10). Написати програму, яка запише в одновимірний масив довжини діагоналей цих прямокутників.

7.48. Масив 6×2 містить: у першому стовпчику швидкості, а другому стовпчику – час руху автомобілів. Написати програму, яка запише в одновимірний масив шлях, пройдений цими автомобілями.

7.49. Масив 4×4 містить випадкові цілі числа у межах від -6 до 6. Написати програму, яка відсортує дані у рядках масиву за спаданням. Метод сортування використати будь-який.

7.50. Масив 4×4 містить випадкові числа у діапазоні $[1;25]$. Написати програму, яка порівняє суми чисел, що містяться на головній і допоміжній діагоналях.

7.51. Масив 3×6 містить випадкові числа у діапазоні від 9 до 19. Написати програму, яка підраховує суму елементів кожного стовпчика масиву і заносить отримані значення в одновимірний масив.

7.52. Масив 7×7 заповнено випадковими числами у діапазоні $[1;40]$. Написати програму, яка буде виконувати пошук у масиві елемента, значення якого задає користувач.

7.53. Масив 4×3 заповнюється з клавіатури. Програма повинна вивести на екран найбільший і найменший елементи масиву.

7.54. Масив 6×2 заповнюється випадковими цілими числами у діапазоні від 1 до 55. Написати програму, яка обчислить суму всіх елементів, що містяться в рядках з парними індексами.

7.55. Масив 6×2 заповнюється випадковими цілими числами у діапазоні від 2 до 20. Написати програму, яка обчислить суму всіх елементів, що містяться в рядках з непарними індексами.

7.56. Масив 3×3 заповнено випадковими цілими числами у діапазоні $[1;12]$. Написати програму, яка замінить місцями перший і останній стовпчики.

7.57. Масив 2×7 заповнено випадковими числами в діапазоні $[1;18]$. Написати програму, яка визначить суму всіх елементів, що знаходяться у стовпчиках з непарними індексами.

7.58. Масив 2×6 заповнено випадковими числами у діапазоні від 1 до 20. Написати програму, яка створить два одновимірні масиви і переписує в них рядки вихідного масиву.

7.59. Масив 3×4 заповнено випадковими числами у діапазоні від 5 до 15. Написати програму, яка перенесе дані по рядках у три одновимірні масиви.

7.60. Масив 3×8 заповнено випадковими числами в діапазоні від -4 до 4. Написати програму, яка виведе на екран номер стовпчика, що містить найбільшу кількість від'ємних чисел.

7.61. Масив 4×6 заповнено випадковими числами в діапазоні від -8 до 8. Написати програму, яка виведе на екран номер рядка, що містить найбільшу кількість додатних чисел.

7.62. Масив 2×12 ініціалізовано на початку програми. Написати програму, яка розіб'є цей масив на два двовимірні масиви, кожен розміром 2×6 .

7.63. Масив 4×4 заповнено випадковими числами у діапазоні від -10 до 10. Написати програму, яка замінить всі від'ємні елементи масиву їх модулями.

7.64. Масив 2×5 заповнено випадковими числами у діапазоні [-3;7]. Написати програму, яка обчислює кількість від'ємних елементів у кожному стовпчику і цю кількість в одновимірній масив.

7.65. Дані про опади в лютому занесено в таблицю 4×7 (кількість тижнів \times кількість днів у тижні). Написати програму, яка виведе на екран номер тижня, в якому кількість опадів найбільша.

7.66. Дані про опади за перші два тижня квітня занесено в таблицю 2×7 (кількість тижнів \times кількість днів у тижні). Написати програму, яка порівняє суми опадів за перший і за другий тиждень.

7.67. Таблиця 3×5 містить випадкові числа в діапазоні [0;40]. Написати програму, яка визначить, скільки чисел діляться на 10.

7.68. Таблиця 5×3 містить випадкові числа у межах від 10 до 99. Програма повинна визначити кількість елементів, які закінчуються на цифру 7.

7.69. Написати програму, яка заповнює таблицю 2×3 степенями числа 6.

7.70. Таблицю 2×5 заповнено степенями числа 3. Написати програму, яка переписує рядки таблиці в два окремі одновимірні масиви.

7.71. У таблицю 2×2 занесено випадкові числа у діапазоні [1;9]. Написати програму, яка розширює масив до розмірів 2×7 і дописує в першому рядку випадкові числа у діапазоні від 10 до 20, у другому рядку – випадкові числа у діапазоні від 21 до 30.

7.72. Таблиця 3×7 містить випадкові числа у діапазоні від 1 до 50. Написати програму, яка виведе номер рядка, сума елементів якого найбільша.

7.73. У масив 6×6 записано випадкові числа у діапазоні від 19 до 55. Програма повинна вивести у рядок перший і третій стовпчики масиву.

7.74. У масив 4×3 занесено випадкові числа у діапазоні від 101 до 200. Програма повинна підрахувати кількість чисел, які закінчуються на цифру 2, у першому і третьому рядках масиву.

7.75. Масив 3×5 містить випадкові числа у діапазоні від -17 до 17. Програма повинна замінити всі від'ємні числа масиву на нулі.

7.76. Масив 6×2 ініціалізується на початку програми. Програма повинна підрахувати кількість елементів, значення яких більші від середнього арифметичного масиву.

7.77. Масив 2×4 заповнюється випадковими числами в діапазоні $[1;20]$. Програма повинна підрахувати кількість елементів, більших від 5 і менших від 10.

7.78. Масив 3×5 заповнено випадковими цілими числами у діапазоні від 1 до 55. Програма повинна поміняти місцями перший і третій стовпчики, другий і четвертий стовпчики.

7.79. Масив розмірами 3×3 заповнюється користувачем. Програма повинна зменшити елементи масиву у F разів. Число F задає користувач.

7.80. Одновимірний масив містить 12 випадкових чисел у діапазоні від 1 до 30. Програма повинна перетворити цей масив у двовимірний масив 3×4 .

7.81. Одновимірний масив містить 12 випадкових чисел у діапазоні від 20 до 50. Програма повинна перетворити цей масив у двовимірний масив 2×6 .

7.82. Масив 3×3 заповнено випадковими цілими числами у діапазоні від 1 до 20. Програма повинна перетворити цей масив в одновимірний і вивести всі елементи масиву у рядок.

7.83. Масив 2×8 заповнено випадковими цілими числами у діапазоні від 101 до 127. Програма повинна перетворити цей масив в одновимірний і вивести всі елементи масиву у рядок.

7.84. У таблицю 3×5 занесено дані про температуру повітря у першій половині вересня (ініціалізовано на початку програми). Програма повинна перетворити масив у одновимірний і відсортувати елементи масиву за зростанням.

7.85. Одновимірний масив заповнено 12 випадковими цілими числами у діапазоні від 1 до 20. Програма повинна перетворити цей масив на двовимірний 2×6 , причому в перший рядок двовимірного масиву слід занести перші 6 елементів початкового масиву, а в другий рядок – другі 6 елементів у зворотному порядку.

7.86. Двовимірний масив 2×4 заповнено випадковими числами у діапазоні від 5 до 25. Написати програму, яка визначить найбільший і найменший елементи масиву.

7.87. Одновимірний масив складається з 12 випадкових чисел у діапазоні від 1 до 50. Написати програму, яка переписе всі дані у двовимірний масив 3×4 , попередньо відсортувавши їх за зростанням.

7.88. Двовимірний масив 3×2 складається з випадкових чисел у діапазоні від -6 до 6. Написати програму, яка переписе всі додатні числа в одновимірний масив.

7.89. Двовимірний масив 3×4 складається з випадкових чисел у діапазоні від 1 до 20. Написати програму, яка переписе всі непарні числа в одновимірний масив А, а парні числа – в одновимірний масив В.

7.90. Двовимірний масив 2×5 містить випадкові числа у діапазоні від 0 до 25. Програма повинна обчислити окремо суму парних елементів і окремо – суму непарних елементів.

7.91. Одновимірний масив складається з 12 випадкових чисел. Програма повинна переписати всі елементи з нього в двовимірний масив 3×4 , замінивши попередньо всі нульові значення на 1.

7.92. Двовимірний масив 3×4 заповнено випадковими цілими числами у діапазоні $[0;120]$. Програма повинна переписати всі елементи масиву, менші за 60 і більші за 0, в одновимірний масив С.

7.93. Двовимірний масив 3×3 заповнено випадковими цілими числами у діапазоні від 0 до 50. Написати програму, яка переписе в одновимірний масив В всі числа, які знаходяться в кутах таблиці.

7.94. Двовимірний масив А розмірністю 4×4 заповнено випадковими числами. Програма повинна скопіювати в одновимірний масив В всі числа, які знаходяться на головній діагоналі, а сам масив А заповнити нулями.

7.95. Двовимірний масив М розмірністю 3×4 заповнено цілими числами, причому перший рядок – випадковими числами у діапазоні $[1;8]$, другий рядок – випадковими числами у діапазоні $[15;22]$, третій рядок – випадковими числами у діапазоні $[101;120]$. Програма повинна обчислити суми елементів кожного стовпчика і вивести їх на екран.

7.96. Двовимірний масив К розмірами 4×5 заповнено випадковими цілими числами у діапазоні $[100; 200]$. Програма повинна створити масив Т таких самих розмірів, заповнений остачами від ділення елементів першого масиву на 7.

7.97. Двовимірний масив $U[3][2]$ заповнено нулями. Написати програму, яка заповнить його степенями числа 4.

7.98. Двовимірний масив $Z[4][4]$ заповнено нулями. Написати програму, яка заповнить його елементами $z[i][j]=i+j$, де i та j – номери рядків і стовпчиків відповідно.

7.99. Двовимірний масив A розмірами 3×5 заповнено випадковими числами. Скопіювати його вміст у масив B так, щоб рядки стали стовпчиками, а стовпчики – рядками.

7.100. Двовимірний масив D розмірами 3×4 заповнено випадковими числами у діапазоні від 1 до 20. Програма повинна перетворити цей масив на масив F розмірами 2×6 , послідовно переписавши у нього елементи масиву D .

Задачі підвищеної складності

7.101. Масив 3×4 заповнено випадковими цілими числами в діапазоні $[10; 55]$. Програма повинна відсортувати середній рядок за зростанням.

7.102. Масив 2×4 заповнено випадковими числами у діапазоні $[20; 90]$. Програма повинна відсортувати дані у кожному стовпчику за зростанням.

7.103. У таблиці 3×4 записані випадкові числа у діапазоні від 1 до 11. Програма повинна перетворити масив на одновимірний і видалити з нього однакові числа.

7.104. Масив 2×6 заповнено випадковими цілими числами у діапазоні $[0; 40]$. Програма повинна відсортувати кожен рядок масиву у порядку зростання.

7.105. Масив 4×4 заповнено випадковими цілими числами у діапазоні $[0; 50]$. Програма повинна замінити всі нульові елементи масиву на степені числа 3.

7.106. Масив 2×2 заповнено випадковими числами (нулями або одиницями). Програма повинна дописати до цього масиву два рядка і два стовпчика, записавши в них степені числа 2.

7.107. Масив A розмірністю 3×3 заповнено випадковими числами у діапазоні $[0; 20]$. Програма повинна переписати у новий масив B , який має розмірність 2×2 , елементи масиву A з такими самими індексами. Всі елементи масиву A , які не скопійовані у масив B , програма повинна переписати в одновимірний масив C і обчислити їх суму.

7.108. Таблицю 3×3 заповнено випадковими цілими числами в діапазоні $[1; 6]$. Написати програму, яка скопіює в одновимірний масив B всі числа, які зустрічаються лише один раз.

7.109. У масив 4×4 записано випадкові числа у межах від 0 до 12. Програма повинна скопіювати в новий одновимірний масив C всі ненульові елементи, які зустрічаються лише один раз.

7.110. Масив 5×5 заповнено випадковими числами у діапазоні $[0;1]$. Програма повинна вивести на екран рядок, який містить найменшу кількість нулів.

7.111. Масив A розмірністю 3×3 заповнено випадковими числами у діапазоні $[0;3]$. Написати програму, яка виведе на екран стовпчик, у якому найбільша кількість нулів.

7.112. Масив K розмірністю 4×4 заповнено випадковими цілими числами у діапазоні від 0 до 25. Написати програму, яка розбиває цей масив на 4 одновимірні масиви відповідно до рядків масиву K .

7.113. Масив A розмірністю 4×4 заповнено випадковими числами у діапазоні від 0 до 4. Програма повинна переписати перший і третій рядки масиву A в одновимірний масив B , а другий і четвертий рядки масиву A – в одновимірний масив C .

7.114. Одновимірний масив A заповнено 12 випадковими цілими числами у діапазоні від 1 до 9. Програма повинна доповнити масив до 16 елементів нульовими значеннями і перетворити масив A на двовимірний масив C розмірністю 4×4 .

7.115. Масив A розмірністю 4×4 містить випадкові числа у діапазоні від 1 до 10. Програма повинна створити чотири масиви розмірністю 2×2 і переписати у них вміст масиву A .

7.116. Масив G має розмірність 4×4 і містить випадкові числа у діапазоні $[0;9]$. Програма повинна переписати більші значення масиву у двовимірний масив A розмірністю 2×4 , а менші значення – у двовимірний масив B розмірністю 2×4 .

7.117. У масивах A, B, C містяться довжини сторін трикутників (випадкові числа у діапазоні від 1 до 5). Кожен масив містить 6 елементів. Програма повинна: 1) перевірити можливість існування трикутників з заданими сторонами; 2) переписати елементи масивів A, B, C у перші три рядки двовимірного масиву Z розмірністю 4×6 ; 3) в останній рядок масиву Z записати 1, якщо трикутник може існувати, і 0, якщо трикутник існувати не може.

7.118. Масив A розмірністю 4×4 містить випадкові числа у діапазоні від 0 до 1. Написати програму, яка замінить у масиві ненульові елементи випадковими числами у діапазоні $[1;25]$.

7.119. Існують два масиви A і B – довільної розмірності, заповнені випадковими числами у діапазоні від 1 до 9. Написати програму, яка запише суму цих масивів у масив C , попередньо перевіривши, щоб масиви A і B мали однакову розмірність.

7.120. Масиви A і B розмірністю 3×3 заповнені випадковими цілими числами у діапазоні від 1 до 7. Написати програму, яка запише добуток цих масивів у масив C .

7.121. Массив Z розмірністю 4×6 заповнено випадковими цілими числами у діапазоні $[1;20]$. Програма повинна обчислити суму всіх елементів, які знаходяться у зовнішніх рядках і стовпчиках масиву.

7.122. Массив C розмірністю 3×3 заповнено випадковими цілими числами у діапазоні від 1 до 7. Написати програму, яка обчислює найбільший елемент з середнього рядка і середнього стовпчика.

7.123. Написати програму, яка заповнить масив M розмірністю 3×4 степенями числа 2 за таким принципом: $M[i][j]=2^{i+j}$.

7.124. Написати програму, яка заповнить масив K розмірністю 2×3 степенями числа 3 за таким принципом: $M[i][j]=3^{i+j}$.

7.125. Массив A розмірністю 4×2 заповнено випадковими числами у діапазоні $[0;25]$. Написати програму, яка перепише перший стовпчик масиву A в одновимірний масив B , а другий стовпчик масиву A – в одновимірний масив C , після чого обчислить суми елементів кожного масиву.

Розділ 8. Списки і кортежі

Списки (list) – це особливий тип даних, який є тільки в мові Python. В інших мовах програмування є лише їх аналоги.

Що ж являють собою списки? Список – це колекція об'єктів певного типу. Вони подібні до масивів, але мають певні відмінності: не настільки типізовані, як масиви, і припускають використання оператора for для обробки елементів списку за їх значеннями, а не за індексами.

Для створення списку служить команда list():

```
x=list('добрий день')
print(x)
print(x[3])
```

На екран буде виведено:

```
['д', 'о', 'б', 'р', 'и', 'й', ' ', 'д', 'е', 'н', 'ь']
р
```

Список може містити будь-яку кількість об'єктів, а може бути порожнім.

Інший спосіб створення списків – використання генератора списків. Генератор списків – це спосіб створити новий список, застосовуючи вираз до кожного елемента послідовності. Генератори подібні до циклу for.

```
c=[c for c in 'good morning']
print (c)
```

На екран буде виведено:

```
['g', 'o', 'o', 'd', ' ', 'm', 'o', 'r', 'n', 'i', 'n', 'g']
```

Для роботи зі списками мова Python має низку функцій та методів (табл.8.1).

Таблиця 8.1

Методи списків

Метод	Характеристика
list.append()	Допишує елемент у кінець списку
list.extend(list2)	Розширює список list, додаючи в кінець елементи списку list2
list.insert(num, x)	Вставляє на позицію num елемент x
list.remove(x)	Видаляє перший елемент у списку, який має значення x. якщо такого елемента немає, то програма виводить повідомлення про помилку
list.pop([n])	Видаляє n-й елемент і повертає його. Якщо індекс не вказано, то видаляється останній елемент

<code>list.index(x,[start[,end]])</code>	Повертає положення першого елемента зі значенням x (при цьому пошук ведеться від позиції start до позиції end)
<code>list.count(x)</code>	Підраховує кількість елементів зі значенням x
<code>list.sort([key=f])</code>	Сортує список відповідно функції f
<code>list.reverse()</code>	Змінює порядок списку на зворотний
<code>list.copy()</code>	Створення копії списку
<code>list.clear()</code>	Очищає список

Методи списків змінюють сам список, тому створювати новий список не потрібно. Іноді для підвищення продуктивності програми застосовуються масиви (ми їх вже розглядали). Для обробки масивів часто застосовують сторонні бібліотеки, наприклад NumPy.

Крім того, для розбиття рядка на слова (розділювач – пробіл) доцільно використати метод `split`:

```
import shlex
sss=input("input text")
words = shlex.split(sss)
```

Для перетворення рядка слів у список слів можна використати функцію `split()`:

```
Spisok=str.split(sss, sep=' ')
```

Для зворотного перетворення списку у рядок слід використати обернену функцію `join`:

```
sss=sss.join(spisok)
```

Приклади задач

Приклад 8.1. Створити список і вивести його на екран у стовпчик.

Розв'язання

```
spisok=[2,7,12,18,24]
for elem in spisok:
    print(elem)
```

Приклад 8.2. Створити список і вивести його на екран у стовпчик. (list02)

Розв'язання

```
import random
a=[]
x=0
for i in range(4):
    x=random.randint(0,100)
    a.append(x)
for elem in a:
```

```
print(elem)
```

Приклад 8.3. Створити список, заповнити його випадковими числами, відсортувати його за зростанням і розвернути у зворотному порядку. (list01)

Розв'язання

```
import math
import random
a=[3]
x=0
for i in range(6):
    x=random.randint(0,18)
    a.append(x)
print(a)
a.sort()
print(a)
b=a
b.reverse()
print(b)
```

Приклад 8.4. Список складається з 8 цілих чисел. Програма повинна вивести перші 4 елементи списку у стовпчик (spisok01).

Розв'язання

```
a=[2,3,2,4,5,3,2,1]
for i in range(0,4,1):
    print(a[i])
```

Приклад 8.5. Користувач вводить кілька слів, розділених пробілами. Програма повинна розбити текст на слова і вивести список цих слів.

```
import shlex
sss=input("input text")
words = shlex.split(sss)
print(words)
```

Завдання для виконання

8.1. Створити список з 10 елементів і визначити, скільки разів у ньому зустрічається символ, введений користувачем.

8.2. Написати програму, яка перетворює рядок, введений користувачем, на список і виводить цей список у стовпчик.

8.3. Написати програму, яка перетворює рядок, створений користувачем, у список, і змінює його порядок на зворотній.

8.4. Написати програму, яка перетворює введене користувачем слово у список і видаляє останній елемент списку.

8.5. Написати програму, яка об'єднує два списки з 3 елементів кожен.

8.6. Список містить 8 цілих чисел. Програма повинна дописати в кінець списку число, введене користувачем.

8.7. Список містить 6 дійсних чисел. Програма повинна дописати до нього 2 числа, введених користувачем.

8.8. Список містить 9 цілих чисел. Програма повинна знайти у списку число, введене користувачем.

8.9. Написати програму, яка зчитує з екрану рядок і перетворює його на список.

8.10. Написати програму, яка зчитує з екрану рядок, перетворює його на список і подвоює цей список.

8.11. Написати програму, яка зчитує з екрану рядок символів, перетворює його на список і виводить на екран один середній символ (якщо кількість елементів списку непарна) або два символи (якщо кількість елементів списку парна).

8.12. Написати програму, яка зчитує з екрану рядок символів і розбиває його на два однакових за довжиною списки.

8.13. Написати програму, яка зчитує з екрану рядок чисел, перетворює його на список і обчислює суму елементів списку.

8.14. Написати програму, яка зчитує з екрану рядок цифр і визначає найбільшу цифру в рядку.

8.15. Написати програму, яка зчитує з рядка рядок цифр і визначає, скільки разів у ньому зустрічається цифра 3.

8.16. Написати програму, яка зчитує з клавіатури рядок цифр, перетворює його в список і дописує в кінець списку першу цифру.

8.17. Написати програму, яка зчитує з екрану рядок символів, перетворює його на список і розташовує елементи списку у зворотному порядку.

8.18. Список цілих чисел ініціалізується на початку програми. Користувач вводить число, яке повинно бути додане на позицію під номером 3. Далі програма повинна підрахувати кількість чисел 11 у списку.

8.19. Список налічує 7 випадкових чисел у діапазоні від 1 до 20. Написати програму, яка визначить, скільки разів у записі чисел зустрічається цифра 2.

8.20. Список налічує 9 випадкових чисел в діапазоні від 1 до 18. Написати програму, яка виводить на екран список у стовпчик у зворотному порядку.

8.21. Список з 18 елементів заповнено випадковими числами в діапазоні [1; 3]. Написати програму, яка видалить зі списку всі значення, що дорівнюють 2.

8.22. Список з 7 елементів заповнений випадковими цілими числами у діапазоні [0; 20]. Написати програму, яка створює другий

список, з 4 випадкових цілих чисел у тому ж діапазоні, і дописує створений список у кінець першого.

8.23. Написати програму, яка надає можливість користувачу дописати у список потрібну кількість цілих чисел, введених за клавіатури. Введення даних припиняється натисканням клавіші 0. Програма повинна також вивести список після завершення введення даних.

8.24. Список містить 12 випадкових чисел в діапазоні від 3 до 30. Написати програму, яка вставляє у список два цілих числа, введених з клавіатури, на позиціях 6 і 2.

8.25. Список з 7 чисел заповнено нулями. Програма повинна змінити значення елементів з непарними індексами на цілі числа, введені користувачем.

8.26. Список налічує 8 чисел і заповнений одиницями. Програма повинна змінити значення елементів списку на такі, які дорівнюють індексам елементів.

8.27. Написати програму, яка створить список з 10 елементів і заповнить його квадратами індексів. Наприклад, елемент `spisok[0]=0`, `spisok[2]=4`.

8.28. Написати програму, яка зчитує з екрану рядок, перетворює його на список, видаляє з нього пробіли і виводить на екран.

8.29. Написати програму, яка зчитує з екрану рядок цифр, перетворює його у список, видаляє з нього всі цифри 9 і виводить на екран.

8.30. Користувач вводить з екрану 10 чисел. Програма повинна записати їх у список і відсортувати у порядку зростання.

8.31. Написати програму, яка зчитує з екрану 10 чисел і формує з них 2 списки (по 5 чисел), виводить їх у рядок кожен окремо.

8.32. Список налічує дані в секундах про проходження дистанції спортсменами і містить 10 елементів. Написати програму, яка виведе на екран 3 найкращі результати.

8.33. Список містить дані про кількість елітних автомобілів у 6 містах і ініціалізується на початку програми. Програма повинна створити дублікат списку і відсортувати його у порядку спадання.

8.34. Список налічує 3 цілих числа. Програма повинна запитати у користувача кілька цілих чисел і дописати їх у кінець списку. Введення даних закінчується введенням числа 0.

8.35. Список містить 10 випадкових чисел у діапазоні від 0 до 27. Програма повинна вивести список у порядку спадання.

8.36. Користувач вводить список довжиною від 4 до 12 елементів. Програма повинна вивести другий і третій елементи.

8.37. Користувач вводить рядок, який складається з цифр. Програма повинна визначити найбільшу цифру і частоту її появи у тексті.

8.38. Написати програму, яка зчитує рядок з екрану, перетворює його на список і дописує між символами пробіли.

8.39. Написати програму, яка замінює всі числа списку на їх модулі.

8.40. Користувач вводить рядок цифр довжиною 8 символів. Програма повинна перетворити їх на список і відсортувати список за зростанням, після чого вивести на екран у стовпчик.

8.41. Користувач вводить рядок цифр довжиною 9 символів. Програма повинна перетворити цей рядок на список і знайти суму елементів списку.

```
rd="123456789"  
a=list(rd)  
b=[]  
for elem in a:  
    b.append(int(elem))  
print(b)  
suma=0  
for elem in b:  
    suma=suma+elem  
print(suma)
```

8.42. Написати програму, яка запитує у користувача рядок символів, перетворює його у список і сортує за зростанням.

8.43. Написати програму, яка заповнює список зі 100 елементів випадковими числами в межах від 0 до 200, сортує список за спаданням і виводить на екран всі числа, більші за 100 (у рядок).

8.44. Користувач вводить межі діапазону випадкових чисел. Програма повинна створити і заповнити 2 списки, кожен з 10 елементів, випадковими числами (довжинами сторін прямокутника). Третій список повинен містити площі цих прямокутників.

8.45. Написати програму, яка створює список, заповнений 6 випадковими числами у діапазоні [1;20], а потім дописує список 3 числами, введеними користувачем.

8.46. Написати програму, яка зчитує елементи списку з екрану, виводить на екран повний список, видаляє з нього останні три елемента і виводить на екран скорочений список.

8.47. Користувач вводить рядок з 10 символів. Програма повинна перетворити його на список, видалити зі списку третій і п'ятий символи, після чого знову вивести список на екран.

8.48. Написати програму, яка зчитує з клавіатури рядок, перетворює його у список і замінює елементи списку з парним індексом на «*».

8.49. Написати програму, яка створює список з 5 випадкових чисел, сортує його за зростанням і дописує до нього симетрично цей же список, відсортований за спаданням.

8.50. Написати програму, яка створює список з 7 випадкових цілих чисел, сортує його за спаданням і виводить у стовпчик елементи з індексами 3 і 5.

8.51. Список містить 8 випадкових цілих чисел. Програма повинна визначити індекс елемента у списку, значення якого введене користувачем.

8.52. Два списки містять по 6 випадкових чисел у діапазоні від 1 до 55. Програма повинна вивести ці списки у два стовпчики поряд.

8.53. Два списки містять по 5 випадкових чисел у діапазоні від 1 до 15. Програма повинна визначити найбільше число в обох списках.

8.54. Два списки містять довжини сторін a і b прямокутників (по 8 випадкових чисел у межах від 3 до 10). Написати програму, яка виведе розміри прямокутника, площа якого найменша.

8.55. Список містить значення радіусів кіл (8 випадкових чисел від 1 до 15). Написати програму, яка обчислить площі і периметри цих кіл і виведе їх на екран у два рядки.

8.56. У список занесено значення температури за останню декаду травня (починаючи з 22 травня) – випадкові числа в діапазоні від 15 до 25°C. Написати програму, яка виведе у два стовпчики числа місяця і відповідні температури.

8.57. Два списки містять набори координат векторів $(x; y)$ на площині. Програма повинна вивести у рядок модулі векторів.

8.58. Список містить дані про популяцію земноводних різних видів (7 випадкових чисел у діапазоні [1000; 15000]). Програма повинна вивести найменшу популяцію.

8.59. Написати програму, яка зчитує рядок символів з екрану, перетворює його в список і перемішує в довільному порядку.

8.60. Вартість зошитів (у грн) занесено в список (довжина списку – 7 елементів). Програма повинна вивести на екран вартість зошитів, більшу за 5 грн.

8.61. Користувач вводить температуру в градусах Цельсія протягом кожного з 7 днів. Програма повинна вивести номер дня, коли температура була найнижча.

8.62. У результаті змагань з ракетомодельного спорту було отримано 8 дійсних чисел – час польоту ракет (їх вводить

користувач). Написати програму, яка обчислить найбільший, найменший і середній час польоту моделей ракет.

8.63. Написати програму, яка зчитує з клавіатури рядок цифр, перетворює його на список цілих чисел і дописує в кінець списку три числа 12.

8.64. Користувач вводить рядок цілих чисел, розділених пробілами. Програма повинна відсортувати ці числа у порядку зростання і вивести список у рядок на екран.

8.65. Користувач вводить рядок цілих чисел, розділених пробілами. Програма повинна визначити, скільки разів у списку зустрічається число, яке стоїть на початку списку.

8.66. Користувач вводить 7 цілих чисел, кожне з нового рядка. Програма повинна вивести суму першого і останнього елементів списку.

8.67. Користувач вводить рядок з 8 цілих чисел, розділених пробілами. Програма повинна видалити зі списку перше і останнє числа.

8.68. Користувач вводить рядок з 7 цілих чисел, відокремлених одне від одного за допомогою пробілів. Програма повинна обчислити суму елементів списку з 3-го до 5-го.

8.69. Дані про кількість відпрацьованих годин понад норму наведено у списку (10 цілих чисел, ініціалізовано на початку програми). Написати програму, яка визначить середній час, переpracьований робітниками.

8.70. Користувач вводить рядок символів англійського алфавіту. Програма повинна відсортувати ці символи у порядку зростання їх порядкового номера в алфавіті і вилучити всі символи «i».

8.71. Написати програму, яка зчитує з екрану чотири слова, введені користувачем, і переставляє місцями перше і друге, третє і четверте слова.

8.72. Список з 10 елементів заповнюється випадковими числами у діапазоні від -10 до +45. Програма повинна вивести на екран у рядок ті елементи списку, які більші за середнє арифметичнє.

8.73. Написати програму, яка видаляє з списку елементи з індексом, заданим користувачем.

8.74. Написати програму, яка видаляє з списку всі елементи із значенням, заданим користувачем.

8.75. Написати програму, яка дописує у початок списку з 7 випадкових чисел 3 числа, заданих користувачем.

8.76. Створити список, який містить 10 додатних чисел (ініціалізується на початку програми). Програма повинна видалити зі списку п'ятий і восьмий елементи.

8.77. Користувач вводить рядок символів. Програма повинна перетворити цей рядок на список і переставити символи у довільному порядку, після чого виконати перетворення списку у рядок і вивести результат на екран.

8.78. Користувач вводить список дійсних чисел, кожне з нового рядка. Програма повинна зменшити кожен елемент списку, що має парний індекс, у 2 рази і вивести список на екран у рядок.

8.79. Написати програму, яка запитує у користувача список цілих чисел, а потім розбиває його на два однакових за довжиною списки і виводить на екран у рядок. Введення даних користувачем закінчується після натискання клавіші 0.

8.80. Створити програму, яка запитує у користувача 10 цілих чисел, дописує їх у кінець списку, обчислює їх суму і виводить на екран.

8.81. Написати програму, яка запитує у користувача 8 двоцифрових чисел, заносить їх у список і переставляє їх у довільному порядку.

8.82. Список з 11 цілих чисел ініціалізується користувачем. Програма повинна визначити індекси найбільшого і найменшого елементів списку.

8.83. Список з 7 цілих чисел заповнюється випадковими числами. Програма повинна обчислити суму найбільшого і найменшого елементів списку.

8.84. Існують 2 списки: перший містить назви фірм, другий – кількість робітників у кожній з фірм. Написати програму, яка виведе на екран у рядок пари значень «назва фірми – кількість робітників».

8.85. Існує список, який складається з 5 цілих чисел. Написати програму, яка запише список у зворотному порядку і допише в кінець списку ці самі значення.

8.86. Написати програму, яка створює список і записує в нього імена учнів класу, які задаються користувачем. Програма повинна відсортувати список за алфавітом і вивести на екран перше і останнє імена.

8.87. Існують два списки, довжиною 7 і 9 елементів, заповнені випадковими числами. Написати програму, яка виведе на екран числа, спільні для обох списків.

```
result = list(filter(lambda elem: elem in b,  
a))
```

8.88. Список з 10 елементів заповнено випадковими числами від 0 до 20. Написати програму, яка виведе на екран частоту, з якою зустрічаються числа 11 і 12.

8.89. Список з 8 елементів заповнено випадковими числами у діапазоні [1;3]. Програма повинна підрахувати, скільки разів у списку зустрічається число 3.

8.90. Два списки, кожен з яких по 6 елементів, заповнено випадковими числами у діапазоні від 1 до 4. Написати програму, яка виведе елементи першого списку, яких немає у другому списку.

```
x = [1,2,3,4,11]
f = [1,11,22,33,44,3,4]

res = list(set(f)-set(x))
print(res)
```

8.91. Два списки, кожен по 10 елементів, заповнені випадковими числами у діапазоні від 1 до 40. Програма повинна видалити з них числа, які зустрічаються в обох списках.

8.92. Два списки, кожен по 8 елементів, заповнено випадковими числами. Програма повинна об'єднати ці списки в один і видалити всі елементи, що мають парні індекси.

8.93. Два списки, кожен по 5 елементів, заповнено випадковими числами у діапазоні від 1 до 20. Написати програму, яка видалить з них усі числа, які наявні в обох списках.

8.94. Два списки, кожен по 8 елементів, заповнені випадковими числами у діапазоні від 10 до 25. Програма повинна залишити в них лише ті елементи, які наявні в обох списках.

8.95. Список містить 12 випадкових чисел у діапазоні від 0 до 20. Написати програму, яка визначає, скільки разів у списку зустрічається кожен елемент.

8.96. Список А складається з 7 цілих чисел, заданих користувачем. Список В з 7 елементів заповнюється випадковими цілими числами. Написати програму, яка поміняє місцями значення списків А і В.

8.97. Список складається з 10 цілих чисел і заповнюється з клавіатури. Програма повинна розбити цей список на два однакових за довжиною і розмістити їх елементи у зворотному порядку.

8.98. Список А складається з 10 випадкових чисел у діапазоні [0;18], список В – з 10 випадкових чисел у діапазоні [3;12]. Написати програму, яка створює список С, кожний елемент якого являє собою суму елементів списків А і В з відповідними індексами.

8.99. Написати програму, яка заповнює список з N елементів випадковими символами англійського алфавіту. Число N задається з клавіатури.

8.100. Користувач заповнює список з N цілих чисел з клавіатури. Програма повинна вивести на екран добуток найбільшого і найменшого елементів списку.

Вкладені списки

Вкладені списки є аналогом двовимірного масиву.

Створення вкладеного списку здійснюється так:

```
a=[[0,0,0],[0,0,0]]
```

У цьому випадку список з 2 рядків 3 стовпчиків заповнюється нулями. Для перебору елементів вкладеного списку слід використати два цикли з лічильником.

Приклад 8.6. Написати програму, в якій створюється і заповнюється вкладений список 3×2 .

Розв'язання

```
import random
a=[[0,0,0],[0,0,0]]
for i in range(2):
    for j in range(3):
        a[i][j]=random.randint(0,20)
print(a[0])
print(a[1])
```

Задачі для виконання

8.101. Створити список 2×4 , який містить довільний набір символів. Програма повинна вивести цей список на екран у вигляді таблиці.

8.102. Розробити програму, яка запитує у користувача цілі числа, заносить їх у список 4×4 , переставляє місцями рядки з парними і непарними індексами і виводить на екран.

8.103. Написати програму, в якій ініціалізується список A 2×4 цілих чисел. Програма повинна переписати в однорядковий список B усі числа, більші за середнє арифметичне, і вивести список B на екран у стовпчик.

8.104. Один з алгоритмів шифрування даних такий: вихідний текст розбивається на групи по 9 символів, кожна група перетворюється на трирядковий список по 3 елементи у кожному. Потім список повертається на 90° , отриманий список знову перетворюється у рядок. Реалізувати цей алгоритм для шифрограми довжиною 9 символів.

8.105. Написати програму, яка заповнює список 3×7 випадковими одноцифровими числами.

8.106. Написати програму, яка заповнює двовимірний список 4×5 випадковими цілими числами у діапазоні $[1;9]$, після чого повертає список на 90° і виводить його порядково на екран.

8.107. Двовимірний список 2×4 заповнюється випадковими числами у діапазоні $[1;20]$. Програма повинна обчислити суму елементів, розміщених по периметру списку.

8.108. Написати програму, яка зчитує з екрану рядок символів, записує його у двовимірний список 5×2 . Зайві символи до списку не потрапляють.

8.109. Користувач заповнює список 4×4 цілими числами. Програма повинна обчислити суму елементів на головній діагоналі і суму елементів на допоміжній діагоналі.

8.110. Написати програму, яка заповнює список 5×5 випадковими числами у діапазоні $[0;125]$, а потім обчислює суму елементів, які знаходяться в останньому стовпчику.

8.111. Написати програму, яка заповнює список 7×7 випадковими числами у діапазоні $[-4;25]$, а потім обчислює суму елементів, які знаходяться в середньому стовпчику.

8.112. Написати програму, яка заповнює список 3×3 випадковими символами англійського алфавіту.

8.113. Написати програму, яка заповнює список 4×4 степенями чисел 2, 3, 4, 5.

8.114. Написати програму, яка заповнює список 5×5 випадковими числами десяткової системи (0 і 1).

8.115. Користувач вводить текст, який має довжину, кратну 16. Програма повинна розбити цей текст на 4 рядки вкладеного списку, потім виводити на екран стовпчики цього списку у рядок.

8.116. Написати програму, яка зчитує з екрану текст, розбиває його на частини довжиною 3 символи і записує посимвольно в багаторядковий список.

8.117. Написати програму, яка заповнює двовимірний список (6 рядків і 3 стовпчики) випадковими числами у діапазоні від 10 до 99, потім перетворює рядки списку у рядки символів і виводить на екран.

8.118. Написати програму, яка заповнює двовимірний список розмірністю 2×4 випадковими символами англійського алфавіту.

8.119. Написати програму, яка заповнює двовимірний список розмірністю 3×6 випадковими символами українського алфавіту.

8.120. Користувач вводить рядок символів довжиною, кратною 3. Програма повинна розбити рядок на вкладений список, який містить 3 стовпчики, а кількість рядків – довільною.

8.121. Користувач вводить рядок тексту довільної довжини. Написати програму, яка запише усі символи рядка у вкладений список розмірністю 5×3 .

8.122. Написати програму, яка заповнює вкладений список 5x3 випадковими символами українського алфавіту, а потім перетворює двовимірний список на рядок.

8.123. Написати програму, яка перетворює рядок, введений користувачем, на вкладений список символів розмірністю 4x2. Зайві символи просто відкидаються. Далі програма повинна додати до кожного елемента списку двоцифрове число і вивести список на екран.

8.124. Вкладений список розмірністю 3x4 містить випадкові символи англійського алфавіту. Написати програму, яка замінить усі елементи середнього рядка списку символами, введеними з клавіатури.

8.125. Користувач заповнює вкладений список розмірністю 3x4 символами. Програма повинна замінити символи першого стовпчика на випадкові символи англійського алфавіту.

Кортежі

Кортеж – це упорядкований набір даних. Цим він подібний до списку. Відмінності полягають у способі і механізмі реалізації, а також у тому, що кортеж не можна змінювати. Після того, як кортеж створено, у нього неможливо внести зміни.

Для створення кортежа використовується вираз
`cc=tuple()`

Якщо дужки порожні, то кортеж буде порожнім. Якщо у дужках записати рядок, то кортеж міститиме символи рядка. Якщо у дужках вказати список, то кортеж буде створено на основі списку.

Інший варіант створення кортежа:

```
ct=tuple("a")
```

У цьому випадку кортеж буде складатися з одного символа.

Створення кортежу на основі списку цілих чисел:

```
ca=tuple([1,2,3])
```

Створення кортежу з рядка даних:

```
cc=tuple("Eagle")
```

Для об'єднання кортежів служить операція +:

```
dd=cc+ca
```

де `cc` і `ca` – два кортежі, створені раніше.

Для підрахунку кількості елементів з вказаним значенням у кортежі служить метод `count`:

```
N=dd.count('E')
```

Для визначення довжини кортежа служить метод `len`:

```
L=len(dd)
```

Отже, основними методами для роботи з кортежами є:

- `index()` – виведення індекса елемента;

- `count()` – підрахунок кількості елементів, що мають задане значення, у кортежі;
- `sum()` – обчислення суми елементів кортежу;
- `max()` – повертає найбільший елемент кортежу;
- `min()` – повертає найменший елемент кортежу;
- `len()` – повертає довжину кортежу.

Основні правила для роботи з кортежами:

- 1) для створення кортежів використовуються метод `tuple()` або порожні дужки;
- 2) елементи кортежу розділяються комами;
- 3) під час створення кортежу слід дотримуватися правил роботи з відповідним типом даних;
- 4) нумерація елементів кортежу може починатися також з кінця кортежу: у такому випадку пишуть номер з мінусом, наприклад: `cc[-2]`;
- 5) кортежі не змінюються, тому застосовуються лише у тому випадку, коли їх змінювати не треба.

Приклади задач

Приклад 8.7. Написати програму, в якій створюється і заповнюється кортеж.

Розв'язання

```
cc=tuple([0,1,2,3,4,5,6,7,8,9,10])
```

Приклад 8.8. Написати програму, в якій створюється кортеж і заповнюється числами від 0 до 10, після чого обчислюється їх сума.

Розв'язання

```
cc=tuple([11,12,13,14,15,16,17,18,19,20])
x=sum(cc)
print(x)
```

Приклад 8.9. Написати програму, в якій рядок символів відсортовується за зростанням, перетворюється на кортеж і виводиться на екран.

Розв'язання

```
yy="hfjfgacergfj"
spisok=list(yy)
spisok.sort()
cf=tuple(spisok)
print(cf)
```

Приклад 8.10. Написати програму, в якій відсортовується кортеж символів.

Розв'язання

```
cc=tuple(["z","a","x","y","m","b","f","t","r"])
spisok=list(cc)
```

```
spisok.sort()
cf=tuple(spisok)
print(cf)
```

Задачі для розв'язання

8.126. Написати програму, яка створює і заповнює кортеж числами від 11 до 20 і обчислює їх суму.

8.127. Написати програму, яка перетворює рядок, введений користувачем, на кортеж, і підраховує кількість літер «а» і «о».

8.128. Кортеж містить дані про температуру повітря за тиждень. Програма повинна обчислити сумарну температуру, середню температуру, кількість днів і вивести на екран у стовпчик усі елементи кортежу.

8.129. Написати програму, яка створює і заповнює кортеж набором випадкових символів англійського алфавіту.

8.130. Написати програму, яка запитує у користувача 2 слова, перетворює на кортежі і об'єднує їх в один великий кортеж.

8.131. Кортеж містить літери англійського алфавіту, які позначають голосні звуки. Програма повинна дописати до кортежу 3 випадкових двоцифрових числа.

8.132. Кортеж містить символи українського алфавіту. Програма повинна підрахувати кількість цих символів у тексті, введеному користувачем.

8.133. Кортеж містить 20 символів. Програма повинна визначити, скільки разів у ньому зустрічається символ, введений користувачем.

8.134. Кортеж містить символи і цілі числа і містить 20 елементів. Програма повинна вивести три перші і три останні символи.

8.135. Кортеж містить символи і складається з 12 елементів. Програма повинна визначити, чи є цей кортеж симетричним.

8.136. Користувач задає кількість елементів і число Z. Програма повинна створити кортеж, заповнений випадковими числами, і перевірити наявність у ньому числа Z.

8.137. Кортеж містить 8 рядків – трьохсимвольних словосполучень. Програма повинна вивести всі елементи кортежу у стовпчик у зворотному порядку.

8.138. Кортеж містить 11 символів. Програма повинна дописати до нього 3 символи, введені користувачем.

8.139. Написати програму, яка об'єднує два кортежі: символи і випадкових чисел, після чого виводить їх на екран.

8.140. Два кортежі містять відповідно довільні символи і цілі числа. Програма повинна об'єднати першу половину першого кортежу і першу половину другого кортежу.

8.141. Три кортежі містять відповідно символи англійського алфавіту, арабські цифри і символи українського алфавіту. Програма повинна дописати до першого кортежу половину другого і половину третього кортежів.

8.142. Написати програму, яка заповнює список символами, введеними користувачем, сортує список і перетворює список на кортеж.

8.143. Кортеж містить довільний набір цілих чисел і створюється на початку програми. Програма повинна відсортувати кортеж за спаданням і вивести утворений кортеж на екран у рядок.

8.144. У кортеж записані 12 літер українського алфавіту у довільному порядку. Програма повинна: 1) визначити, чи наявні у кортежі літери «а» і «к»; 2) відсортувати кортеж за спаданням і вивести його на екран у стовпчик.

8.145. Два кортежі містять відповідно символи англійського алфавіту і десяткові цифри. Кожен кортеж складається з 10 елементів. Програма повинна відсортувати кортежі за спаданням і об'єднати їх у третій кортеж.

8.146. Написати програму, яка виконує пошук у кортежі символа, введеного користувачем.

8.147. Кортеж містить 7 довільних символів англійського алфавіту. Програма повинна дописати у початок кортежу 5 символів, введених користувачем.

8.148. У кортеж записано 11 символів англійського алфавіту. Програма повинна видалити з кортежу символ, введений користувачем.

8.149. Кортеж містить 5 цілих чисел. Програма повинна дописати у середину кортежу 3 цілих числа, введених користувачем.

8.150. Кортеж містить 6 символів українського алфавіту. Програма повинна відсортувати його за зростанням і дописати до нього цей же кортеж, відсортований за спаданням.

Розділ 9. Словники

Короткі теоретичні відомості

Словники є складним типом даних, який властивий лише мові Python. Словник – це аналог списку і множини водночас. Словник являє собою невпорядкований набір елементів. Кожен елемент словника має ключ, який однозначно ідентифікує елемент у словнику (як індекс елемента в списку). Але індекси елементів списку – цілі числа, які являють собою неперервну послідовність і зростають від першого елемента до останнього. Ключі у словнику не обов'язково числа – вони можуть бути рядками або списками.

Для створення словника використовується функція `dict()`. В дужках можна записати вирази типу `ключ=значення`, кожна така пара відокремлюється комою від інших. Функції можна передати список, кожним елементом якого також є списки.

Якщо дужки в функції `dict()` залишити порожніми, створюється порожній словник. Якщо в дужках записати назву існуючого словника, то буде створено його копію. Для вкладених елементів виконується копіювання не самих елементів, а посилань на них.

Словник можна створити, якщо у фігурних дужках вказати вирази типу `ключ=значення`, розділивши ці пари комами. Між ключем і значенням пишеться двокрапка.

Звернення до елемента словника здійснюється так само, як і до елемента списку. Спочатку пишеться ім'я словника, а у квадратних дужках – ключ. Наприклад:

```
z=slovník[key]
```

Такі є самі конструкції використовуються для того, щоб змінити значення елемента словника.

Основні методи словників:

`dict.clear()` – очищає словник;

`dict.copy()` – повертає копію словника;

`classmethod dict.fromkeys(seq[, value])` – створює словник з ключами з `seq` і значенням `value` (за замовчуванням `None`);

`dict.get(key[, default])` – повертає значення ключа, а якщо його немає, повертає `default` (за замовчуванням `None`);

`dict.items()` – повертає пари (ключ, значення);

`dict.keys()` – повертає ключі словника;

`dict.pop(key[, default])` – видаляє ключ і повертає значення.

Якщо ключа немає, повертає `default` (за замовчуванням генерує виняток);

`dict.popitem()` – видаляє і повертає пару (ключ, значення).

Якщо словник, порожній, повертає виняток `KeyError`;

`dict.setdefault(key[, default])` – повертає значення ключа, а якщо його немає, то створює ключ зі значенням `default` (за замовчуванням `None`);

`dict.update([other])` – оновлює словник, додаючи пари «ключ-значення» з `other`. Існуючі ключі перезаписуються. Повертає `None`;

`dict.values()` – повертає значення у словнику.

Приклади задач

Приклад 9.1. Написати програму, яка створює список співробітників фірми з даними про їх вік і виводить список на екран.

Розв'язання

```
workers=dict()
workers['Петренко']='22'
workers['Васильків']='24'
workers['Галюк']='31'
workers['Сивков']='38'
workers['Черенцький']='29'
workers['Смолінчук']='24'
workers['Сергеев']='41'
for key in workers:
    print (key+' '+workers[key])
```

Приклад 9.2. Написати програму, яка виконує пошук даних у словнику за прізвищем робітника.

Розв'язання

```
workers=dict()
workers['Петренко']='22'
workers['Васильків']='24'
workers['Галюк']='31'
workers['Сивков']='38'
workers['Черенцький']='29'
workers['Смолінчук']='24'
workers['Сергеев']='41'
for key in workers:
    print (key)
z=input("Введіть прізвище для пошуку")
for key in workers:
    if key==z:
        print (key+' '+workers[key])
```

Приклад 9.3

Розв'язання

Приклад 9.4

Розв'язання

Приклад 9.5

Написати програму, яка виводить населення міст зі списку (дані про населення містяться у словнику).

Програма

```
cities= dict()
cities['London'] = '9200'
cities['Paris'] = '8700'
cities['Rome'] = '6300'
spisok = ['London', 'Rome', 'Paris', 'Tokyo']
for elem in spisok:
    if elem in cities:
        print('The population of ' + elem + ': ' + cities[elem])
    else:
        print('it is absent :'+elem)
```

Завдання

9.1. Створити словник, у якому наведено назви хімічних елементів і їх позначення у таблиці Менделєєва. Структура словника – <назва>: <позначення>. Написати програму, що виводить позначення елемента за його назвою.

9.2. Створити словник, у якому наведено назви хімічних елементів і їх позначення у таблиці Менделєєва. Структура словника – <позначення>: <назва>. Написати програму, що виводить назву елемента за його позначенням.

9.3. Створити словник, у якому наведено назви хімічних елементів і їх позначення у таблиці Менделєєва. Структура словника – <позначення>: <назва>. Написати програму, що визначає наявність у словнику відповідного хімічного елемента.

9.4. Створити словник, у якому назви держав співставляються з назвами їх столиць. Кількість елементів словника – 5. Програма повинна виводити столицю держави, назву якої вводить користувач.

9.5. Словник містить перелік учнів класу і їх зріст (всього – 8 елементів). Програма повинна виводити зріст учня, а користувач вводить прізвище.

9.6. Словник містить назви 5 європейських країн і їх площі. Програма повинна виводити площу країни за її назвою.

9.7. Словник містить дані про населення 7 міст України. Програма повинна виводити населення міста, коли користувач вводить назву міста.

9.8. Словник містить назви і відносні атомні маси 5 хімічних елементів (на ваш вибір). Написати програму, яка буде виводити відносну атомну масу хімічного елемента за його назвою.

9.9. Розробити словник, який міститиме назви 6 міст України і їх населення. Програма повинна виводити населення міста за його назвою.

9.10. Розробити словник, який міститиме імена і дані про середні оцінки 4 учнів. Програма повинна виводити оцінку учня за його іменем.

9.11. Розробити словник, який міститиме прізвища 5 учнів класу і їх зріст у см. Програма повинна вивести зріст учня за його прізвищем.

9.12. Створити словник, який містить назви 7 фірм і їх оборотний капітал. Програма повинна вивести значення оборотного капіталу за назвою фірми.

9.13. Розробити словник, який містить 5 назв сушвантажних суден, їх тоннаж і порт приписки. Програма повинна вивести дані про судно за його назвою.

9.14. Розробити словник, який містить 10 назв танкерів і країн, яким вони належать. Програма повинна вивести назви танкерів за країною-власником.

9.15. Створити словник, який складається з 7 марок автомобілів і відповідних країн. Програма повинна виводити всі марки автомобілів за назвою країни, заданою користувачем.

9.16. Створити словник, який містить 7 позначень класів (наприклад, 8-А, 8-Б тощо) і чисельність учнів у цих класах. Програма повинна вивести назви класів, чисельність яких більша за число x , задане користувачем.

9.17. Написати словник, який містить назви міст і країни їх розміщення (10 елементів). Написати програму, яка виводить: 1) країну, якій належить місто (користувач вводить назву міста); 2) міста, які належать країні (користувач вводить країну).

9.18. Створити словник, який містить назви фірм-виробників автомобілів (8 назв) і країни, в яких вони створені. Програма повинна виконати пошук за назвою країни і за назвою фірми.

9.19. Розробити словник, який міститиме назви країн і назви континентів, до яких належать ці країни. Програма повинна вивести всі країни, що належать континенту, назву якого обирає користувач.

9.20. Написати словник, який містить назви і площі 9 країн Європи (Велика Британія, Ірландія, Іспанія, Франція, Швейцарія, Італія, Німеччина, Нідерланди, Бельгія). Розробити програму, яка буде виводити усі назви країн, площа яких більша заданої користувачем.

9.21. Створити словник, який містить дані про 5 міст Київської області: назву і площу. Програма повинна створити копію словника і вивести на екран назву і населення міста, назву якого задав користувач.

9.22. Написати програму, яка містить словник, що складається з номерів груп і їх чисельності (всього 9 груп). Програма повинна виводити: 1) сумарну чисельність груп; 2) номер найбільшої групи; 3) номер найменшої групи. Відповідно програма повинна містити меню для вибору користувачем потрібної операції.

9.23. Написати програму, що містить словник, який складається з назв країн Північної Америки і їх площ. Програма повинна: 1) виводити площу країни за назвою; 2) виводити 2 країни, площа яких найбільша. У програмі повинно бути меню, за допомогою якого користувач вибирає відповідну функцію.

9.24. Розробити 2 словники, які містять відомості про назву, населення і площу 5 країн Південно-Східної Азії (на вибір). Програма повинна вивести на екран: 1) назву і населення найбільшої за площею країни; 2) назву і площу країни, населення якої найменше. Програма повинна містити меню, за допомогою якого користувач має можливість вибрати потрібну функцію.

9.25. Написати програму, компонентом якої є словник, що містить дані: річка – довжина (для 10 річок). Програма повинна: 1) вивести на екран увесь зміст словника; 2) дописати у словник дані для 2 річок.

9.26. Написати програму, у процесі роботи якої користувач заповнює словник даними учнів: прізвище зріст. Програма повинна вивести прізвище учня, зріст якого найбільший.

9.27. Створити програму, яка містить словник, дані у який додаються користувачем. Словник повинен складатися з полів: <назва птаха> : <континент, на якому мешкає цей птах>. Після завершення введення даних програма повинна вивести всі записи на екран.

9.28. Написати програму, яка містить словник. Словник повинен складатися з записів з полями: назва дерева – максимальна висота – континент, на якому росте дерево.

9.29. Написати програму, яка містить словник, що складається з 4 записів, кожен яких поділяється на поля: назва острова – океан. Програма повинна виконувати пошук по назвам океанів.

9.30. Створити словник, який містить записи про 4 обласні центри України (Чернігів, Київ, Суми, Полтава): назва міста – населення – площа. Програма повинна: 1) створити копію словника і вивести її на екран; 2) запитати у користувача дані ще про одне місто і записати їх у другий список; 3) вивести перший список на екран.

9.31. Розробити словник, який містить такі поля: ідентифікатор, марка автомобіля, кількість пройдених кілометрів. Написати програму, яка запитує дані у користувача і заносить їх у словник, а потім виводить список автомобілів на екран.

9.32. Розробити словник, який містить такі дані про 8 підприємств: назву, місто. Програма повинна створити копію словника і видалити з копії списку останні два записи.

9.33. Написати програму, яка містить словник з відомостями про номери та розміри земельних ділянок. Програма повинна дописати у словник дані, введені користувачем, і вивести словник на екран.

9.34. Написати програму, яка містить словник з відомостями про площу п'яти областей України: Київська, Чернігівська, Житомирська, Черкаська, Вінницька. Список містить такі поля: назва області і площа. Програма повинна вивести на екран всі елементи словника.

9.35. Розробити програму, яка використовує словник з відомостями про міста і їх населення. Програма повинна виконувати 2 функції: додавання нових елементів у словник і виведення всього словника на екран.

9.36. Написати програму, яка використовує словник з відомостями населення країн Південної Африки: ПАР, Зімбабве, Ангола, Мозамбик, Намібія. Програма повинна виводити відомості про країну за її назвою.

9.37. Написати програму, яка використовує словник з відомостями про країни Тихоокеанського регіону: Австралію, Нову Зеландію, Самоа, Папуа-Нову Гвінею. Програма повинна за назвою країни виводити її площу і населення.

9.38. Написати програму, яка створює словник з відомостями про країни: назва країни, площа. Програма повинна запитувати дані у користувача і записувати її у словник.

9.39. Написати програму, яка надає користувачу можливість створити словник з 10 записів. Словник повинен містити такі поля: ключ, прізвище учня, зріст учня. Програма повинна визначити максимальний і мінімальний зріст, середній зріст усіх учнів.

9.40. Створити словник, що містить 7 записів. Кожен запис складається з 2 полів: прізвище учня, оцінка учня з алгебри (оцінки учнів за 12-бальною системою). Програма повинна вивести на екран прізвища 3 учнів, які мають найкращі оцінки з алгебри.

9.41. Розробити програму, яка створює словник з 5 записів з відомостями про вантажопідйомність автомобілів різних марок. Кожен запис має містити поля: ключ, марка автомобіля,

вантажопідйомність. Програма повинна вивести марку автомобіля, який має найбільшу вантажопідйомність.

9.42. Словник містить дані про співробітників фірми: прізвище та ініціали, рік початку роботи на фірмі. Програма повинна вивести дані про двох співробітників, які працюють на фірмі найдовше.

9.43. Словник містить дані про співробітників фірми: прізвище та ініціали, рік народження. Програма повинна вивести дані про трьох співробітників, які мають найменший вік.

9.44. Створити словник, який міститиме відомості про автомобілі, які належать фірмі. Кожен запис словника повинен складатися з 2 полів: марка автомобіля, рік останнього капітального ремонту. Програма повинна вивести дані про автомобілі, з часу проходження капітального ремонту якими пройшло понад 8 років.

9.45. Розробити програму, яка міститиме словник з 9 записів з відомостями про мотоцикли. Кожен запис повинен містити поля: марка мотоцикла, максимальна швидкість. Програма повинна виводити максимальну швидкість мотоцикла за його маркою.

9.46. Словник містить 11 записів про характеристики різних металів. Кожен запис складається з таких полів: назва металу, густина металу. Програма повинна вивести на екран усі відомості про ці метали.

9.47. Написати словник, який міститиме дані про результати змагань спортсменів з бігу. Кожен запис повинен містити поля: прізвище, час подолання дистанції у секундах. Програма повинна вивести прізвища спортсменів, які зайняли перші три місця.

9.48. Словник містить дані про печиво різної марки. Кожен запис має такі поля: марка печива, ціна 1 кг. Написати програму, яка обчислить вартість ящика печива всіх марок печива, якщо маса ящика становить 2,5 кг.

9.49. Словник містить характеристики масла різних сортів. Кожен запис повинен складатися з таких полів: сорт масла, ціна у гривнях. Програма повинна вивести назви найдорожчого і найдешевшого сортів масла.

9.50. Написати, програму, яка використовує словник з 10 записів. Ключами для записів є назви країн, а значеннями – назви столиць цих країн.

Задачі підвищеної складності

9.51. Написати програму, яка використовує словник з 10 записів типу <країна>:<столиця>. Написати програму, яка перетворить цей словник у інший, типу <столиця>:<країна>.

9.52. Словник А містить 10 записів про столиці країн Європи. Масив В містить 10 назв столиць. Програма повинна підрахувати,

скільки назв столиць співпадають для масиву і словника. У програмі використати бібліотеку `numpy`.

9.53. Існують словники з назвами міст. Кожен запис у цих словниках ключовим полем має назву міста, а значенням – назву країни, якій належить місто. Програма повинна визначити, скільки назв міст є спільними для обох словників.

9.54. Існує словник з назвами міст (10-15 записів). Кожен запис являє собою відношення <назва міста>:<назва країни>. Написати програму, яка підраховує кількість міст, які належать кожній країні.

9.55. Існує словник з прізвищами учнів, кожен запис якого являє собою відношення <прізвище>:<клас>. Написати програму, яка виведе на екран усі записи, які відносяться до учнів певного класу. Клас задає користувач.

9.56. Словник містить дані про гірські вершини у форматі <назва вершини>:<висота вершини>. Написати програму, яка виведе назви усіх вершин, висота яких більша за середнє арифметичне усіх вершин.

9.57. У словнику записані дані про 7 озер України. Кожен запис має формат <назва озера>:<площа поверхні озера>. Програма повинна вивести на екран назви озер, що мають найбільшу і найменшу площі.

9.58. Словник містить дані про 5 областей України. Записи мають формат <назва області>:<площа області>. Програма повинна вивести на екран у рядок площі областей, упорядковані за спаданням.

9.59. Написати програму, яка надає можливість користувачу заповнити словник з записами у форматі: <назва міста>:<населення міста>. Програма повинна вивести на екран назви двох міст, які мають найбільше населення.

9.60. Розробити програму, яка надасть користувачу можливість заповнити словник даними про назви фірм і кількість працівників у них. Програма повинна: 1) обчислити загальну кількість працівників; 2) обчислити середню кількість працівників; 3) вивести назву фірми, яка має найбільше працівників.

9.61. Словник містить дані про міста Чернігівської області у форматі <назва міста>:<населення>. Програма повинна вивести назви 3 міст, які мають найменше населення.

9.62. Словник містить індекси поштового зв'язку 7 уявних міст у форматі <індекс>:<назва міста> (назви міст визначити самостійно). Кожен індекс формується за принципом: перші дві цифри повинні бути «99», наступні три цифри – випадкове трицифрове число. Програма повинна вивести на екран усі написи, відсортувавши назви міст за спаданням.

9.63. У словнику написані дані про рік випуску 7 моделей легкових автомобілів. Програма повинна вивести дані 2 автомобілів, вік яких найбільший.

9.64. Написати програму, яка використовує 2 словники. Перший словник містить дані про сорт печива і ціну 1 кг цього печива. Другий словник містить дані про сорт печива і масу ящика цього печива. Програма повинна вивести на екран вартість ящиків печива.

9.65. Дані про дерев'яні бруски записані у 3 словника. Перший словник містить відомості про номер бруска і сорт деревини, другий словник – номер бруска і густину, третій словник – номер бруска і об'єм. Написати програму, яка виведе відомості про номер бруска, матеріал і масу.

9.66. Дані про співробітників фірми занесено у два словники. Перший словник містить дані про прізвище і рік народження. Другий словник містить дані про прізвище і рік початку роботи в цій фірмі. Програма повинна вивести на екран всі дані про найстаршого працівника і співробітника, який працює найдовше в цій фірмі.

9.67. Дані про 7 областей України записані у 2 словники. Перший словник містить дані у форматі <назва області>:<площа області>. Другий словник містить відомості у форматі <назва області>:<населення області>. Програма повинна вивести на екран відомості про назву області і густину населення.

9.68. Відомості про 5 європейських країн написано у два словники: перший містить дані про назву країни і населення, другий містить дані про назву країни і її площу. Написати програму, яка виведе на екран назви країн, які мають найбільшу і найменшу густину населення.

9.69. Перший словник містить відомості про номер ділянки і зібраний урожай ріпаку. Другий словник містить дані про номер ділянки і її площу. Номер ділянки складається з 2 літер англійського алфавіту і 3 цифр. Написати програму, яка обчислить середню урожайність з цих ділянок.

9.70. Відомості про результати роботи 8 фірм зведено у 2 словники. Перший словник містить записи у форматі <назва фірми>:<кількість людино-днів>. Другий словник містить записи у форматі <назва фірми>:<отриманий прибуток>. Програма повинна вивести усі дані про фірму, яка виявила найбільшу ефективність роботи.

9.71. Відомості про жорсткі диски записані у 3 словники. Перший словник містить записи у форматі <умовне позначення>:<марка диска>. Другий словник містить записи у форматі <умовне позначення>:<обсяг диска>. Третій словник

містить записи у форматі <умовне позначення>:<вартість диска>. Програма повинна вивести усі дані диска, для якого ціна збереження 1 ГБ даних найменша.

9.72. Словник містить дані про назви програмних продуктів і фірм-розробників. Програма повинна створити другий словник, назви програмних продуктів взяти з першого, а їх ціни запитати у користувача.

9.73. Написати програму, яка створює два словники, а користувачу надає можливість заповнити їх. Перший словник має містити записи у форматі <назва міста>:<рік заснування>, а другий словник – записи у форматі <назва міста>:<населення>. Програма повинна вивести на екран назви двох міст – найстарішого і того, що має найбільше населення.

9.74. Написати програму, яка містить словник з таблицею густин різних пород деревини (словник створюється на початку програми і там же заповнюється даними). У програмі створити меню, за допомогою якого користувач може виконувати такі дії: 1) пошук густини деревини за назвою породи; 2) додавання запису у кінці словника; 3) видалення запису за значенням ключа, введеним користувачем.

9.75. Словник містить 12 записів про столиці країн у форматі <назва>:<рік заснування>. Програма повинна вивести на екран всі відомості про столиці країн, які містяться у списку spisok1.

Розділ 10. Підпрограми

Короткі теоретичні відомості

У мові програмування Python можна створювати підпрограми – функції і процедури.

Процедури виконують одну і ту ж послідовність дій і не повертають ніяких значень. Процедура задається так:

```
def <назва> (параметр1, параметр 2...):  
    <команди>
```

Функція повертає одне значення в основну програму.

Для задання функції використовується набір команд:

```
def <назва функції> (параметр1, параметр2...):  
    <набір команд>  
    return <значення>
```

Приклад підпрограмипроцедури, яка обчислює середнє арифметичне:

```
def medium(a, b):  
    return (a + b) / 2  
print(medium(a, b))
```

Для того, щоб викликати процедуру в Python, необхідно записати її назву і в дужках – значення параметрів або змінних, які передаються в процедуру.

Наприклад:

```
medium(5,2)
```

Усі змінні у мові Python поділяються на глобальні локальні. Локальні використовуються лише всередині окремої підпрограми. Для того, щоб зробити змінну глобальною, слід виконати команду:

```
global z
```

Де z – ім'я змінної. Ця команда визначає змінну як глобальну. Змінну можна використовувати в підпрограмі, при цьому її значення буде змінюватися і в основній програмі.

Приклади задач

Приклад 10.1. Користувач вводить кількість елементів масиву. Підпрограма повинна заповнити масив випадковими цілими числами в діапазоні [0;120] і вивести масив на екран.

```
import random  
from array import *  
def printmasiv(n):  
    masiv=array('i',[])  
    for i in range (n):  
        x=random.randint (0,120)  
        masiv.append(x)  
    print(masiv)  
z=int(input("введіть кількість елементів масиву"))
```

```
printmasiv(z)
```

Приклад 10.2. Написати програму, яка виводить на екран двовимірний масив. Для створення масиву використати відповідну функцію.

```
import random
import numpy as np
def make_masiv():
    massive1=np.ones((3,2),dtype=np.int)
    for i in range(3):
        for j in range (2):
            massive1[i][j]=random.randint(0,10)
    return(massive1)
miy=make_masiv()
print(miy)
```

Завдання для виконання

Процедури

10.1. Написати підпрограму, яка виводить на екран N рядків, в яких повторюються 10 символів *.

10.2. Написати підпрограму, яка виводить на екран заданий символ задану кількість разів.

10.3. Написати підпрограму, яка виводить на екран задану кількість символів &.

10.4. Написати програму, яка виводить на екран рядок символів ! задану кількість разів. Кількість рядків і кількість символів у рядку задаються користувачем.

10.5. Написати програму, яка виводить рамку з символів =. Ширина і висота рамки задаються користувачем.

10.6. Написати програму, яка виводить на екран піраміду з символів %. Кількість рядків задає користувач.

10.7. Написати програму, яка містить процедуру для обчислення катета прямокутного трикутника. Користувач задає гіпотенузу і другий катет.

10.8. Користувач вводить два цілих числа – розміри прямокутника. Підпрограма повинна намалювати на екрані прямокутник з символів «*» заданих розмірів.

10.9. Написати програму для розв'язання квадратного рівняння. Для цього написати процедуру обчислення і виведення на екран коренів квадратного рівняння.

10.10. Користувач вводить кількість елементів масиву. Підпрограма повинна заповнити масив випадковими цілими числами в діапазоні [0;120] і вивести масив на екран.

10.11. Написати програму, яка заповнює масив $M \times N$ випадковими цілими числами у діапазоні $[0; 45]$. Для цього створити відповідну процедуру.

10.12. Користувач вводить масив з 7 цілих чисел. Програма повинна визначити середнє арифметичне масиву і його найбільший елемент. Для цього створити відповідну процедуру.

10.13. Користувач вводить три числа a, b, c – три ребра прямокутного паралелепіпеда. Створити процедуру, яка обчислює діагональ паралелепіпеда і діагоналі трьох неоднакових сторін.

10.14. Написати програму, яка малює три прямокутника розмірами axb у точці з координатами (x, y) (за допомогою модуля turtle). Для цього написати процедуру для малювання прямокутника.

10.15. Написати програму, яка малює правильний шестикутник, який складається з правильних трикутників. Для цього створити процедуру, яка малює правильний трикутник. Вхідний параметр: сторона трикутника. У програмі використати модуль turtle.

10.16. Правильний шестикутник вписано в коло радіусом R . Написати процедуру, яка обчислює і виводить на екран площу і периметр шестикутника.

10.17. Написати програму, що містить процедуру, яка обчислює проєкції вектора на осі Ox та Oy . Вхідні дані: координати двох точок – початку і кінця вектора.

10.18. Користувач вводить координати початку і кінця вектора. Процедура повинна обчислити координати вектора і його довжину.

10.19. Написати програму, яка малює задану кількість горизонтальних ліній заданої довжини і заданої товщини. Для цього написати процедуру малювання однієї лінії. Вхідні параметри: довжина і товщина лінії. У програмі використати модуль turtle.

10.20. Написати програму, яка малює задану кількість правильних трикутників. Для цього створити процедуру, яка малює один правильний трикутник. Вхідні параметри: сторона трикутника, координати стартової точки. У програмі використати модуль turtle.

10.21. Написати програму для малювання ромбів заданої кількості і заданих розмірів. Для цього створити процедуру, яка малює ромб з діагоналями заданих розмірів, нижня вершина якого знаходиться у точці з заданими координатами. У програмі використати модуль turtle.

10.22. Написати процедуру, яка малює паралелограм з сторонами 70 і 50, кут між сторонами 40° . Вхідні параметри для процедури: координати x і y лівої нижньої вершини паралелограма. Одна з сторін паралелограма паралельна осі x . У програмі використати модуль turtle.

10.23. Написати програму, яка зображає задану кількість трапецій. Для цього створити процедуру, яка малює рівнобічну трапецію зі сторонами 80, 40, 30 і 30. У програмі використати модуль turtle.

10.24. Написати програму, яка обчислює об'єм і площу поверхні кулі і виводить їх на екран. Для цього створити відповідну процедуру.

10.25. Написати процедуру для малювання рамки по периметру вікна. Використати модуль turtle.

10.26. Написати процедуру, яка малює прямокутник 200x300 і заштриховує його тонкими чорними вертикальними лініями. Використати модуль turtle.

10.27. Написати підпрограму, яка малює прямокутник 250x250 і заштриховує його горизонтальними синіми лініями. Використати модуль turtle.

10.28. Написати процедуру, яка виводить на екран квадрат $Y \times Y$ символів «z». Число Y вводиться з клавіатури.

10.29. Написати процедуру, яка виводить на екран суцільний прямокутник розмірами $A \times B$ різних символів. У рядках з парним номером вивести символи "R", у рядках з непарним номером – символи "F".

10.30. Написати програму, яка виводить на екран порожнистий прямокутник символів «*». Розміри прямокутника задає користувач.

10.31. Написати підпрограму, яка виводить на екран піраміду символів "&" висотою h і шириною b . Числа a і b задаються з клавіатури.

10.32. Написати підпрограму, яка виводить на екран рівнобедрений трикутник висотою h і довжиною основи a , заповнений символами «*».

10.33. Написати підпрограму, яка малює на екран ромб з символів «*» висотою h і шириною b . Величини b , h вводять користувач.

10.34. Написати підпрограму, яка малює на екрані правильний шестикутник зі стороною a . Величину a задає користувач. У програмі використати модуль turtle.

10.35. Написати підпрограму, яка малює на екрані вертикально розташований ромб з гострим кутом 29 і стороною a . Значення a вводиться з клавіатури. Використати модуль turtle.

10.36. Написати підпрограму, яка обчислює і виводить на екран об'єм і площу бічної поверхні циліндра. Вхідні параметри: радіус циліндра і його висота.

10.37. Написати підпрограму, яка обчислює і виводить на екран об'єм і площу поверхні кулі. Вхідний параметр: радіус кулі.

10.38. Написати програму, яка обчислює і виводить на екран об'єм і площу бічної поверхні конуса. Вхідні параметри: радіус і висота конуса.

10.39. Написати програму, яка малює на екрані 6 кіл з радіусом, який зменшується. Для цього створити підпрограму, яка малює на екрані коло радіусом R , починаючи з точки з координатами (x, y) . У програмі використати модуль turtle.

10.40. Написати програму, яка малює на екрані коло з двома діаметрами. Для цього створити підпрограму, яка малює коло і два діаметра. Вхідні параметри: радіус кола, координати початкової точки. У програмі використати модуль turtle.

10.41. Написати програму, яка використовує головне меню, в якому користувач може вибрати одну з дій: 1) малювання квадрата заданих розмірів з початком у заданій точці; 2) малювання кола заданого радіуса з початком у заданій точці. Для цього створити 2 підпрограми, аргументи для яких: координати точки $A(x,y)$, з якої починається малювання фігури і R (розмір фігури – радіус або сторона квадрата). У програмі використати модуль turtle.

10.42. Написати програму, яка містить процедуру для шифрування даних. Процедура приймає рядок символів, міняє місцями символи з парними і непарними індексами і виводить результат на екран.

10.43. Створити програму для шифрування даних. Для цього розробити процедуру, яка розбиває рядок символів, введений користувачем, на буквосполучення по 2 символи і міняє місцями буквосполучення з парними і непарними номерами.

10.44. Розробити програму для шифрування даних. Для цього створити процедуру, яка розбиває рядок на трьохсимвольні буквосполучення, а потім міняє місцями буквосполучення з парними і непарними номерами.

10.45. Написати процедуру, яка дешифрує повідомлення, а саме – видаляє коден другий символ з рядка, введеного користувачем. Вхідний аргумент – рядок тексту, результат повинен виводитися на екран.

10.46. Написати програму, яка містить процедуру, яка приймає два рядки тексту і видаляє з них символи, які є спільними в обох рядках. Результат роботи – два рядки – виводяться на екран.

10.47. Написати програму, яка малює правильний п'ятикутник. Для цього створити процедуру, аргументами якої є довжина сторони п'ятикутника і координати x, y початкової точки. Використати модуль turtle.

10.48. Створити програму, яка обчислює периметр і площу прямокутника. Для цього розробити процедуру, аргументами якої є числа a і b – сторони прямокутника.

10.49. Написати програму, яка малює правильний восьмикутник. Для цього розробити і підключити до основної програми відповідну підпрограму. Аргументи: сторона восьмикутника, координати початкової точки. У програмі використати модуль turtle.

10.50. Написати програму, яка малює літеру К великих розмірів. Для цього розробити і підключити до основної програми відповідну підпрограму. Аргументи: координати початкової точки x і y . У програмі використати модуль turtle.

10.51. Написати програму, яка виводить на екран індекси першого і останнього входження символа «a» у рядку, введеному користувачем. Для цього створити і викликати в основній програмі відповідну процедуру.

10.52. Створити процедури виведення елементів списку з парними і непарними індексами на екран і викликати ці процедури у тексті основної програми. У програмі заповнити список з 20 елементів випадковими цілими числами у діапазоні $[1;40]$, а для виведення на екран використати відповідні підпрограми.

10.53. Масиви $A[]$ і $B[]$ містять відповідно верхній і нижній діапазони випадкових чисел. Написати програму, яка обчислює ці випадкові числа і виводить їх на екран у вигляді списку. Для цього створити відповідну процедуру.

10.54. Одна з систем кодування передбачає заміну символа числом. Простіший аналог такої системи передбачає заміну перших 10 символів англійського алфавіту цифрами від 0 до 9. Створити процедуру, яка замінює символи рядка, введеного користувачем, на цифри і виводить їх на екран.

10.55. Написати програму, яка малює у кутах вікна квадрати, зафарбовані червоним. Сторона такого квадрата – 100 пікселів. Для цього створити процедуру, яка малює квадрат у точці, обчислений програмою. Вхідні параметри: координати x і y початкової точки малювання квадрата. У програмі використати модуль turtle.

10.56. Написати програму, яка заштриховує вікно програми синіми лініями під кутом 45° з кроком d . Для цього створити підпрограму, яка малює одну лінію.

10.57. Масиви $A[]$ і $B[]$ містять розміри прямокутників і являють собою випадкові числа у діапазоні від 1 до 9. Написати програму, яка виведе на екран номери і площі прямокутників, площа яких більша від 25. Для цього створити процедуру, яка обчислює площу прямокутника і виводить її на екран.

10.58. Написати програму для обробки масиву довільної довжини, заповненого випадковими числами у діапазоні $[-6;10]$. Для цього розробити процедуру, яка виводить на екран кількість нульових елементів у масиві. Підпрограма повинна отримувати аргументи: масив і його довжину.

10.59. Написати програму, яка перевіряє належність точок колу радіусом R з центром у початку координат. Двовимірний масив A містить 2×10 випадкових чисел – координати точок. Створити процедуру, яка виводить на екран координати точок, які належать колу.

10.60. Написати програму, яка заповнює масив 3×4 випадковими числами і виводить на екран вміст цього масиву. Для виведення на екран масиву створити процедуру, яка приймає 2 параметри: масив і кількість елементів у ньому.

10.61. Написати програму, яка перемножує всі елементи масиву на довільне число і виводить результат на екран. Масив 2×3 заповнюється випадковими числами. Для множення елементів масиву на число і виведення результатів на екран створити відповідну процедуру. Використати бібліотеку NumPy.

10.62. Написати програму, яка створює два масиви 2×3 , заповнює їх випадковими числами і перемножує їх елементи. Для множення масивів і виведення результату на екран створити відповідну процедуру. Використати бібліотеку NumPy.

10.63. Написати програму, яка створює два масиви 3×3 , заповнює їх випадковими числами (у діапазоні відповідно $[0;10]$ і $[20;30]$), додає масиви і виводить результат на екран. Для додавання масивів і виведення на екран розробити процедуру, яка приймає два аргументи – масиви, над яким слід виконати дії. Використати бібліотеку NumPy.

10.64. Масив 3×4 заповнено випадковими числами у діапазоні від 0 до 20. Написати програму, яка перетворить цей масив у однорядковий і вилучить з масиву всі елементи, які дорівнюють нулю. Для перетворення масиву в одновимірний і видалення з масиву елементів з нульовим значенням створити процедуру. У програмі використати бібліотеку NumPy.

10.65. Масив 4×4 містить випадкові цілі числа у діапазоні $[-2;10]$. Написати програму, яка обчислює кількість нульових елементів і заміняє їх на число 11. Розробити процедуру, яка підраховує кількість нульових елементів масиву, заміняє їх на число 11 і виводить масив на екран. Для роботи з двовимірними масивами використати бібліотеку NumPy.

10.66. Масив 2×3 містить випадкові цілі числа у діапазоні від -4 до 8. Написати програму, яка заміняє елементи масиву з від'ємним

значенням на їх модулі. Вивести на екран масив до обробки і після обробки. Для виведення масиву на екран створити відповідну процедуру. У програмі використати бібліотеку NumPy.

10.67. Написати програму, яка малює на екрані 2×3 рядки квадратів, розмір яких задано користувачем. Сторони квадратів повинні бути паралельні осям координат. Для малювання квадрата створити процедуру, яка приймає один аргумент – сторону квадрата. У програмі використати модуль turtle.

10.68. Написати програму, яка малює N рядків квадратів, у кожному з яких 4 квадрати. Розміри квадрата і число N задає користувач. Для малювання квадрата створити відповідну процедуру, яка приймає один аргумент – довжину сторони квадрата. У програмі використати модуль turtle.

10.69. Написати програму, яка малює на екран M рядів по N квадратів у кожному. Числа M і N задає користувач. Для малювання квадрата зі стороною 70 пікселів створити відповідну процедуру. Для малювання фігур підключити модуль turtle.

10.70. Написати програму, яка малює на екрані N рівносторонніх трикутників, розміщених в один ряд так, щоб їх основи були паралельні осі Ox . Для малювання рівностороннього трикутника створити процедуру, яка приймає один аргумент – довжину сторони трикутника. У програмі використати модуль turtle.

10.71. Написати програму, яка малює на екрані 2 ряди по 4 рівносторонні трикутники у кожному. Основи трикутників повинні бути паралельні осі Ox . Для цього створити процедуру, яка малює рівносторонній трикутник з стороною 60 пікселів. Для малювання фігур підключити модуль turtle.

10.72. Написати програму, яка малює на екрані 3 ряди по 3 рівносторонніх трикутника у кожному так, щоб їх основи були розміщені горизонтально. Для малювання рівностороннього трикутника зі стороною a створити процедуру, яка приймає один аргумент – довжину сторони a . Для малювання трикутників підключити модуль turtle.

10.73. Розробити програму, яка намалює на екрані ряд з 4 кіл радіусом 70 пікселів, причому сусідні кола повинні дотикатися. Для малювання кола і переміщення пензля створити процедуру. У програмі підключити модуль turtle.

10.74. Написати програму, яка малюватиме 2 ряди по 3 кола у кожному. Радіус кола задає користувач. Процедура повинна переміщувати пензль у потрібну точку і малювати коло заданого радіуса. Вхідні параметри: координати точки, з якої необхідно почати малювання кола. Для малювання фігур підключити модуль turtle.

10.75. Користувач задає кількість квадратів, які повинні бути розташовані в один ряд, що буде розтягнутий на всю ширину вікна. Довжину сторони квадрата повинна розраховувати програма. Процедура повинна малювати квадрат заданих розмірів. Для малювання фігур підключити модуль turtle.

10.76. Масив з N елементів складається з випадкових чисел у діапазоні [0;100]. Написати програму, яка міститиме такі процедури: 1) для обчислення середнього арифметичного елементів масиву; 2) для обчислення різниці між найбільшим і найменшим елементами масиву. Кожна з процедур повинна вивести результат на екран. У самій програмі створити меню, за допомогою якого користувач може вибрати потрібну операцію з масивом.

10.77. Користувач вводить рядок тексту довжиною від 10 до 100 символів. Написати підпрограми, які повинні: 1) розбити текст на групи по 2 символи і записати в масив; 2) розбити текст на групи по 4 символи і записати в масив. Якщо довжина рядка не кратна числам 2 або 4, кожна з підпрограм повинна дописати в кінці рядка нулі. Кожна з підпрограм повинна вивести результат на екран. В основній програмі створити меню, за допомогою якого користувач матиме можливість вибрати потрібну опцію. Для роботи з масивами символів використати бібліотеку numpy.

10.78. Написати програму, яка здійснюватиме шифрування повідомлень за таким принципом: 1) спочатку текст повідомлення доповнюється символами до тих пір, поки довжина тексту не стане кратною 4; 2) текст розбивається на групи по 4 символи; 3) кожна група символів змінює свій порядок на зворотній; 4) отримані групи символів склеюються в один рядок і виводяться на екран. Для цього створити одну процедуру, яка і виконуватиме всі потрібні дії. Для створення і обробки масивів символів використати бібліотеку numpy.

10.79. Написати програму, яка буде виводити на екран рядок випадкових символів заданої довжини N. Для цього створити 2 процедури: 1) виводить на екран рядок випадкових символів англійського алфавіту; 2) виводить на екран рядок випадкових символів українського алфавіту. У програмі створити меню, за допомогою якого користувач може вибрати потрібну операцію.

10.80. Написати програму, яка розіб'є введений рядок тексту на слова. Для цього створити процедуру, яка розбиває рядок на масив слів і виводить цей масив на екран у стовпчик. Для обробки масиву рядків використати бібліотеку numpy.

10.81. Написати програму, яка намалює прямокутник і вписаний у нього ромб, вершини якого лежать на серединах сторін прямокутника. Для цього створити дві процедури. У програмі використати модуль turtle.

10.82. Написати програму, яка малює на екрані ромб, вершини якого лежать на серединах сторін вікна. Кожний наступний ромб повинен бути подібним до нього, а діагоналі – меншими на 40 пікселів. Для малювання ромба створити процедуру. У програмі використати модуль turtle.

10.83. Написати програму, яка малює восьмикутник, чотири сторони якого лежать на сторонах вікна. Для цього створити відповідну процедуру. Для малювання фігури використати модуль turtle.

10.84. Розробити програму, яка малює квадрат і його діагоналі. Користувач повинен задати розміри квадрата. Для малювання фігури створити процедуру. У програмі використати модуль turtle.

10.85. Написати програму, яка малює N ромбів в один ряд так, щоб їх діагоналі були паралельні осям координат. Для малювання одного ромба створити процедуру. Для малювання фігур підключити до програми модуль turtle.

10.86. Розробити програму, яка шифрує повідомлення, яке складається з цифр. Шифрування здійснюється у такій послідовності: текст повідомлення розбивається на двоцифрові числа, між ними вставляється текст у вигляді випадкових двоцифрових чисел, після чого частини повідомлення склеюються в один рядок і виводяться на екран. Для роботи з масивом рядків використати бібліотеку numpy.

10.87. Написати програму для дешифрування повідомлень, зашифрованих за принципом, вказаним у попередній задачі. Для цього створити процедуру, яка прийматиме рядок повідомлення, розбиватиме його на групи символів по 4 цифри, видаляє з кожної з таких груп останні дві цифри і склеює частини в один рядок. Для роботи з масивом рядків використати бібліотеку numpy.

10.88. Написати програму, яка малюватиме схематичне зображення будинку. Для цього створити дві процедури: 1) малює дах у вигляді трикутника; 2) малює стіни у вигляді прямокутника. Використати модуль turtle.

10.89. Написати програму, яка малює на екрані прямокутні рівнобедрені трикутники, розташовані в один ряд. Для цього створити процедуру, яка малює прямокутний рівнобедрений трикутник так, щоб катети були паралельні осям координат. Катети трикутника задаються користувачем. У програмі використати модуль turtle.

10.90. Написати програму, яка малює горизонтальний ряд кіл радіусом 60 пікселів. Для цього створити процедуру, яка малює коло з першим відступом від початкової точки. Для малювання фігур використати модуль turtle.

10.91. Користувач задає координати початкової точки для малювання п'ятикутника. Розробити процедуру, яка намалює правильний п'ятикутник зі стороною 120 пікселів з початком у цій точці. У програмі використати модуль turtle.

10.92. Користувач задає чисельник і знаменник звичайного дробу. Процедура повинна перевірити, чи можна цей дріб перетворити у десятковий, і, якщо це так, то вивести на екран утворений десятковий дріб.

10.93. Масив складається з 6 рядків, кожен рядок містить по 4 символи. Написати програму, яка виведе на екран по 2 символи у стовпчик. Для цього створити окрему процедуру. У програмі використати модуль numpy.

10.94. Масив складається з 10 рядків тексту, кожен рядок по 6 символів. Програма повинна вивести на екран усі рядки масиву, що містять літеру «а». Для цього створити відповідну процедуру. У програмі використати модуль numpy.

10.95. Масив містить 7 рядків тексту. Написати програму, яка виводить на екран рядки, що містять лише цифри. Для цього створити відповідну процедуру. До програми підключити модуль numpy.

10.96. Масив складається з 11 випадкових чисел у діапазоні від 10 до 100. Програма повинна вивести на екран усі прості числа. Для цього створити процедуру.

10.97. Масив складається з 12 випадкових чисел у діапазоні [1;99]. Програма повинна дописати до кожного числа випадковий символ англійського алфавіту і вивести масив на екран у стовпчик. У програмі використати модуль numpy.

10.98. Користувач вводить три числа x_1 , x_2 і x_3 . Процедура повинна згенерувати три випадкові числа a , b , c , які являють собою випадкові числа відповідно в діапазонах $[1;x_1]$, $[1;x_2]$, $[1;x_3]$, а потім обчислити об'єм прямокутного паралелепіпеда з ребрами a, b, c і вивести його на екран.

10.99. Один із способів дешифрування повідомлення полягає у наступному: з рядка тексту зашифрованого повідомлення копіюються символи, індекси яких є простими числами, і таким чином утворюється дешифроване повідомлення.

10.100. Словник містить дані про населення 4 країн Далекого Сходу: Японію, Китай, В'єтнам, Лаос. Написати процедури: 1) для виведення на екран запису про конкретну країну; 2) додання запису у словник.

Функції

10.101. Створити функції для обчислення об'єму і площі поверхні куба. Вхідний аргумент – довжина ребра куба a .

Використати у програмі ці функції для обчислення об'єма і площі поверхні куба. Довжина ребра куба задається користувачем.

10.102. Розробити функції для обчислення об'єму і площі поверхні кулі. Вхідний аргумент – радіус кулі. Використати ці функції у програмі для виконання відповідних обчислень.

10.103. Створити функції для обчислення об'єму і площі бічної поверхні конуса. Аргументи для функцій – радіус кола (основи конуса) і висота конуса. Використати ці функції у програмі для виконання відповідних обчислень.

10.104. Написати функції для обчислення площі поверхні і об'єму прямокутного паралелепіпеда. Аргументи: довжини ребер прямокутного паралелепіпеда. Написати програму, яка порівнюватиме об'єми двох паралелепіпедів і площі поверхні двох паралелепіпедів.

10.105. Написати функції для обчислення сум елементів масиву з парними і непарними індексами. Вхідні дані для функцій: масив і число (кількість елементів масиву). Написати програму, яка генерує масив випадкових цілих чисел і виконує обчислення за функціями.

10.106. Написати програму для обчислення периметра і площі ромба. Для цього написати дві функції, аргументи для яких – довжини двох діагоналей.

10.107. Написати програму, що містить дві функції: 1) для обчислення площі прямокутника; 2) для обчислення периметра прямокутника. Вхідні параметри для обох сторін: сторони a , b прямокутника. Програма повинна використовувати ці функції для виконання відповідних обчислень.

10.108. Написати програму, яка містить дві функції: 1) для обчислення периметра квадрата; 2) для обчислення площі квадрата. Вхідний параметр один для обох функцій – сторона квадрата. Програма повинна обчислювати або периметр, або площу квадрата (на вибір користувача).

10.109. Правильний шестикутник вписано у коло радіусом R . Написати функції для обчислення площі і периметра шестикутника. Вхідний параметр один – радіус описаного кола R . Написати код основної програми, щоб вона могла обчислювати площу або периметр за вибором користувача.

10.110. Написати програму, що містить дві функції, які обчислюють відповідно проекції вектора на осі Ox і Oy . Аргументи для функцій: чотири дійсних числа (координати двох точок – початку і кінця вектора).

10.111. Написати програму, яка визначає окремо координату x вектора і координату y вектора. Для цього написати дві окремі

функції. Вхідні параметри для обох функцій – чотири числа (координати x , y початку і кінця вектора).

10.112. Написати програму для обчислення скалярного добутку двох векторів. Для цього у програмі створити відповідну функцію, яка приймає чотири параметри: координати x та y обох векторів.

10.113. Користувач вводить два чотиризначних числа. Програма повинна визначити число, сума цифр якого більша. Для цього слід створити функцію, яка визначає суму цифр чотиризначного числа.

10.114. Користувач вводить три тризначні числа. Програма повинна визначити, у якого з цих чисел більша середня цифра. Для цього створити функцію, яка визначає середню цифру числа.

10.115. Написати програму, яка обчислює суму коренів квадратних рівнянь. Використати 2 функції обчислення коренів квадратного рівняння.

10.116. Написати підпрограму для обчислення периметра трикутника за його трьома сторонами.

10.117. Написати підпрограму для обчислення периметра прямокутного трикутника. Параметри підпрограми – 2 довжини катетів.

10.118. Написати програму для порівняння периметрів двох прямокутних трикутників. У програмі використати підпрограму для обчислення периметра прямокутного трикутника за його двома катетами.

10.119. Написати програму для порівняння площ двох прямокутних трикутників. У програмі використати підпрограму для обчислення площі прямокутного трикутника за його двома катетами.

10.120. Написати програму для порівняння площ двох ромбів. У програмі використати функцію для обчислення площі ромба за довжинами його діагоналей.

10.121. Написати програму для обчислення периметра трикутника. У програмі використати підпрограму для перевірки умови існування трикутника.

10.122. Написати програму для знаходження суми площ N рівнобедрених трикутників. Для цього створити функцію обчислення площі рівнобедреного трикутника за довжиною його бічної сторони і кута між ними.

10.123. Написати програму, яка наповнює масив площами рівнобічних трапецій. Для цього використати функцію обчислення площі рівнобічної трапеції за його основою і бічною стороною.

10.124. Користувач вводить довжини основ і бічної сторони рівнобічної трапеції. Програма повинна заносити площі трапецій у масив. Використати підпрограму обчислення площі трапеції.

10.125. Трикутник задано координатами вершин. Написати програму для обчислення периметра трикутника. Використати функцію обчислення відстані між двома точками на площині.

10.126. Написати програму для обчислення суми площ двох трикутників. Використати функцію обчислення площі рівнобедреного прямокутного трикутника за довжиною його катета.

10.127. Написати програму для порівняння об'ємів двох куль. У програмі використати функцію обчислення об'єму кулі за її радіусом.

10.128. Написати програму для визначення найбільшого з трьох чисел. У програмі використати функцію визначення більшого з двох чисел.

10.129. Написати програму для визначення найбільшого з чотирьох чисел. У програмі використати функцію визначення більшого з двох чисел.

10.130. Написати програму для обчислення периметра прямокутного трикутника. Використати функцію для обчислення периметра прямокутного трикутника. Аргументи функції – дві довжини катетів.

10.131. Написати програму для визначення того, чи є слово паліндромом. У програмі використати функцію для визначення того, чи є слово паліндромом.

10.132. Користувач вводить два натуральних числа. Написати програму, яка визначає число, в якого сума цифр більша. Використати функцію обчислення суми цифр числа.

10.133. Написати програму для розв'язання квадратного рівняння. Для цього створити функцію обчислення дискримінанта квадратного рівняння.

10.134. Користувач вводить з клавіатури масив, який налічує 8 цілих чисел. Написати програму для обчислення дільників елементів масиву, для цього створити відповідну функцію.

10.135. Користувач вводить з клавіатури масив з 5 цілих чисел. Програма повинна записати в інший масив найбільші дільники цих чисел. Для цього створити відповідну функцію.

10.136. Написати програму, яка заповнює масив з 10 чисел об'ємами куль. Для цього створити функцію, яка обчислює об'єм кулі. Вхідний параметр – радіус кулі.

10.137. Поле, засіяне кукурудзою, має форму круга радіусом R. Написати функцію, яка обчислює урожай, зібраний з поля. Аргументи функції – радіус поля і урожайність з 1 га.

10.138. Поле, засіяне житом, має форму ромба. Програма повинна обчислити загальний урожай, зібраний з поля. Для цього написати функцію, аргументи якої – дві діагоналі ромба і урожайність з 1 га.

10.139. Написати програму, яка обчислює урожайність культури. Для цього створити функцію, яка приймає два вхідних параметри – площу поля і масу зібраного урожаю.

10.140. Написати програму, яка обчислює урожай картоплі, зібраний з двох полів. Для цього створити функцію, що обчислює урожай картоплі з одного поля. Вхідні параметри: довжина і ширина поля, що має форму прямокутника. Урожайність картоплі з'ясувати самостійно і записати у функції у вигляді константи.

10.141. Написати програму, яка визначає кількість води, потрібну для поливу трьох полів. Для визначення кількості води, потрібної для поливу одного поля, створити функцію з такими вхідними аргументами: довжина і ширина поля, висота шару води (їх задає користувач, а потім вони переносяться у функцію у вигляді аргументів).

10.142. Написати програму, яка визначає кількість фарби для фарбування бічних сторін бака, що має форму прямокутного паралелепіпеда. Для цього створити функцію, що обчислює площу бічної поверхні бака. Аргументи функції: довжини сторін основи a, b , висота h бака. Питома витрата фарби на 1 м^2 – 180 г. Результат повинен бути виведений у кілограмах.

10.143. Написати програму, яка обчислює сторони двох квадратів. Для цього створити функцію, яка обчислює сторону квадрата за довжиною діагоналі квадрата.

10.144. Написати функцію, яка шифрує рядок тексту таким чином: між символами вхідного тексту вставляються довільні символи англійського алфавіту.

10.145. Створити програму, яка містить функцію для визначення найменшої цифри натурального числа довільної довжини.

10.146. Створити програму, яка містить функцію, що переставляє місцями першу і останню цифри чотирицифрового числа.

10.147. Розробити програму, яка містить функцію, що обчислює добуток двох середніх цифр чотирицифрового числа.

10.148. Написати програму, яка містить функцію для обчислення добутку середніх цифр двох трицифрових чисел. Вхідні параметри функції: два трицифрових числа.

10.149. Створити програму, яка обчислює довжину висоти трикутника, проведену до основи. Для цього написати функцію, яка обчислює площу трикутника за трьома його сторонами.

10.150. Існують два квадрати, вписані у кола різного радіуса. Написати програму, яка порівняє площі цих квадратів. Для цього створити функцію, яка обчислює площу квадрата за значенням радіуса описаного кола.

10.151. Написати програму, яка обчислює найбільший спільний дільник трьох чисел. Для цього створити функцію для визначення найбільшого спільного дільника двох цілих чисел. Аргументи для функції: два цілих числа. У програмі записати виклик цієї функції.

10.152. Написати програму, яка обчислює найбільший спільний дільник чотирьох чисел. Для цього створити функцію для визначення найбільшого спільного дільника двох цілих чисел. Аргументи для функції: два цілих числа. У програмі записати виклик цієї функції.

10.153. Написати програму, яка визначає індекси останнього включення символу в рядках, введених користувачем. Для цього створити функцію, яка визначає індекс останнього входження заданого символу у рядок. Аргументи функції: символ і рядок. Результат: індекс символу у рядку.

10.154. Написати програму, яка підраховує кількість нульових елементів у двовимірному масиві. Для цього створити функцію, яка заповнює двовимірний масив випадковими числами у діапазоні [0;24]. Аргументи функції: числа M , N – розмірність масиву.

10.155. Створити для порівняння суми цифр цілих чисел, введених користувачем. Для цього розробити і викликати в основній програмі функцію для обчислення суми цифр числа (розмір числа не визначений).

10.156. Написати програму, яка виводить на екран N рядків, заповнених випадковими цілими числами. Для цього розробити функцію `makeList()`, яка заповнює список з 10 елементів випадковими числами у діапазоні від 0 до 20 і перетворює його в рядок.

10.157. Написати програму для обчислення периметра трикутника, заданого координатами вершин. Для цього розробити функцію, яка визначатиме відстань між двома точками на площині. Аргументи: координати двох точок.

10.158. Написати програму для обчислення периметра чотирикутника, заданого координатами вершин. Для цього розробити функцію, яка визначатиме відстань між двома точками на площині. Аргументи: координати двох точок.

10.159. Написати програму, яка обчислює три висоти трикутника. Користувач задає довжини a, b, c сторін трикутника. Розробити і викликати функції: 1) для обчислення площі трикутника за трьома сторонами (аргументи – сторони трикутника); 2) для обчислення висоти, проведеної до основи трикутника (аргументи – площа і основа).

10.160. Написати програму для обробки масиву. Для цього створити функцію, яка підраховує кількість нульових елементів масиву, і функцію, яка підраховує кількість ненульових елементів масиву. Аргументи для обох функцій – масив і його довжина.

10.161. Написати програму, яка обчислює загальну площу земельних угідь, подану у вигляді масиву. Для цього створити функцію, яка заповнює масив випадковими числами у діапазоні $[0;100]$. Вхідний параметр – число N (кількість елементів масиву).

10.162. Написати програму для шифрування повідомлень. Для цього створити функцію, яка вписує по два випадкових символа після кожного символа вхідного повідомлення.

10.163. Написати програму для обчислення площі трапеції. Для цього створити функцію для обчислення середньої лінії трапеції.

10.164. Написати програму, яка заповнює одновимірний масив з 8 елементів площами прямокутників. Для цього створити функцію, яка обчислює площу одного прямокутника. Аргументи – довжина і ширина прямокутника.

10.165. Написати програму, яка заповнює одновимірний масив з 7 елементів значеннями об'єму прямокутних паралелепіпедів. Для цього розробити функцію обчислення об'єму прямокутного паралелепіпеда за його трьома ребрами.

10.166. Два масиви $A[]$ і $B[]$ містять по 10 випадкових чисел (у діапазоні від 2 до 7) – катети прямокутних трикутників. Написати програму, яка заносить у масив $C[]$ площі цих трикутників. Для цього створити функцію обчислення площі прямокутного трикутника за його катетами.

10.167. Два масиви $Ro[]$ і $V[]$ містять по 12 випадкових чисел – відповідно об'єми і густини тіл. Написати програму, яка запише у масив $M[]$ маси цих тіл. Для цього створити функцію, яка обчислює масу тіла за його об'ємом і густиною.

10.168. Написати програму, яка заповнить масив $Ro[]$ густинами тіл (випадкові числа у діапазоні від 0,4 до 8,9), а масив $V[]$ – об'ємами цих тіл (випадкові числа у діапазоні від 0,1 до 10). Для цього написати 2 функції: 1) для заповнення масиву густин випадковими числами; 2) для заповнення масиву об'єми випадковими числами.

10.169. Масиви A[] і B[] містять по 9 елементів (випадкові числа у діапазоні від 1 до 9), кожний з елементів являє відповідно довжину і ширину прямокутника. Написати програму, яка визначить найбільший і найменший периметри прямокутників. Для цього створити функцію, яка обчислює периметр прямокутника за його сторонами.

10.170. Таблиця містить 12 чисел (випадкові числа у діапазоні від 0,1 до 2) – радіуси циліндрів. Висота усіх циліндрів 1,7. Програма повинна обчислити суму об'ємів цих циліндрів. Для цього створити функцію, яка обчислює об'єм одного циліндра (вхідний параметр – радіус циліндра).

10.171. Масиви R1[] і R2[] містять відповідно зовнішній і внутрішній радіуси кільця. Написати програму, яка обчислить площі цих кілець і виведе на екран найбільшу площу. Для цього створити підпрограму обчислення площі кільця.

10.172. Написати програму, яка визначає кількість літер, які позначають голосні звуки, у тексті, введеному користувачем. Для цього створити функцію, яка обчислює кількість таких літер в одному рядку. Користувач має ввести 3-7 рядків.

10.173. Написати програму, яка перетворює введений текст на масив чисел, шифруючи таким чином початкове повідомлення. Замість кожної літери масив повинен отримати номер її за алфавітом. Для визначення номера літери а алфавіті створити функцію.

10.174. Цистерна для нафти має форму циліндра діаметром D і висотою h. На фарбування 1 м² поверхні потрібно k грамів фарби. Написати програму, яка обчислить кількість фарби, потрібну для фарбування цистерни зсередини. Для цього створити функцію, яка обчислює площу поверхні, яку треба пофарбувати. Вхідні параметри: D, h.

10.175. Написати програму для генерування серійних номерів. Принцип утворення серійного номера: перший символ – випадкова літера англійського алфавіту. Для цього дописується через дефіс випадкове трицифрове число. Для генерування випадкової літери англійського алфавіту створити функцію.

10.176. Написати програму для створення рядка випадкових символів заданої довжини. Для цього створити функцію, яка генерує випадковий символ українського алфавіту. Кількість символів у рядку задається користувачем.

10.177. Написати програму для генерування масиву випадкових номерів телефону одного з операторів. Для цього слід створити функцію, яка створює телефонний номер. Телефонний номер має починатися символами «+38066», після чого пишеться випадкове семизначне число. Основна програма повинна вивести на

екран у стовпчик усі створені номери. У програмі використати бібліотеку `numpy`.

10.178. Написати програму, яка генеруватиме випадковий номер автомобіля. Користувач вводить серію (2-3 символи англійського алфавіту), а функція повинна прийняти цей рядок, згенерувати випадкове чотирицифрове число, дописати до нього серію і повернути створений номер в основну програму.

10.179. Написати програму, яка перевірятиме, чи є в масиві однакові серійні номери програмних продуктів. Масив має налічувати 5-10 серійних номерів, утворених за допомогою функції, написаної власноруч. Функція генерує номер у вигляді рядка формату “sss-*nnn*”, де *sss* – серія програмного продукту (наприклад, ZQN), а *nnn* випадкове тризначне число. Для створення масиву використати бібліотеку `numpy`.

10.180. Написати програму, яка створює масив з 10 рядків (по 4 символи), причому кожен рядок містить випадкові символи англійського алфавіту. Для створення рядка випадкових символів розробити відповідну функцію. Для роботи з масивом рядків підключити модуль `numpy`.

10.181. Написати програму, яка упорядковує масив номерів авіарейсів, який складається з 12 елементів. Для цього створити функцію, яка генерує номери авіарейсів. Номер авіарейсу складається з двобуквенного виразу (наприклад, “NH”), після якого записується двозначне число. У програмі використати бібліотеку `numpy`.

10.182. Написати програму, яка упорядковує масив номерів автобусних рейсів. Номер рейсу складається з двох великих літер англійського алфавіту і випадкового трицифрового числа. Для створення номера розробити функцію. У програмі використати бібліотеку `numpy`.

10.183. Серійний номер програмного продукту формується за таким принципом: “LLL-NNN-LLL-NNN”, де LLL – три випадкових великих літери англійського алфавіту, а NNN – випадкове тризначне число. Для створення серійного номера розробити відповідну функцію. Програма повинна заповнити масив 12 випадковими серійними номерами і вивести їх у стовпчик на екран. Для роботи з масивами використати бібліотеку `numpy`.

10.184. Один із принципів шифрування повідомлень такий: символи початкового повідомлення записуються на позиціях, індекси яких є простими числами. Написати програму, яка шифруватиме повідомлення за цим принципом, створивши відповідну функцію.

10.185. Написати програму, яка порівняє потенціальну енергію двох тіл, маси і висоти яких задано користувачем. Для цього створити функцію, яка обчислює потенціальну енергію тіла.

10.186. Написати програму, яка записує у масив $E[12]$ кінетичну енергію тіл. Маси тіл і їх швидкості містяться у масивах відповідно $M[]$ і $V[]$. Для обчислення кінетичної енергії створити відповідну функцію.

10.187. Написати програму, яка обчислює суму відстаней від точки до початку координат усіх точок, координати яких записані у масив $M[][]$. Для цього створити функцію, яка обчислюватиме відстань від точки до початку координат. Для роботи з двовимірним масивом використати бібліотеку `numpy`.

10.188. Написати програму, яка обчислюватиме суму відстаней точок від початку координат. Координати точок записуються у двовимірний масив, у якому перший стовпчик містить координати x , а другий стовпчик – координати y . У програмі використати дві функції: 1) створює і наповнює масив координат точок випадковими числами від 0 до 10; 2) обчислює відстань від конкретної точки до початку координат.

10.189. Відомості про металеві заготовки записані у два масиви: масив $RO[]$ містить густини металів (г/см^3 , випадкові числа від 6,3 до 10,2), масив $V[]$ містить об'єми тіл (у см^3 , випадкові числа від 1 до 16). Програма повинна обчислити суму мас у сіх заготовок. Для цього створити функції: 1) для створення і наповнення масиву $RO[]$; 2) для створення і наповнення масиву $V[]$; 3) для створення і заповнення масиву $M[]$, який міститиме маси заготовок.

10.190. Два масиви містять дані про пробіг автомобілів за 1 день і витрату палива за цей же час. Програма повинна визначити найменшу і найбільшу витрату палива на 1 км. Для цього створити і підключити 2 функції: 1) для створення і наповнення масиву $S[]$ даними про пробіг машин – випадковими числами у діапазоні від 80 до 300; 2) для створення і наповнення масиву $V[]$ даними про витрату палива автомобілями за день – випадкові числа у діапазоні від 30 до 100.

10.191. Емулятор ігрового автомату працює таким чином: автомат генерує три випадкових літери англійського алфавіту, якщо вони однакові, то це означає виграш. Написати програму, яка генерує набір з трьох випадкових великих літер англійського алфавіту до тих пір, поки не співпадіть всі три літери, і підраховує кількість утворених комбінацій. У програмі створити і використати дві функції: 1) генерування випадкового символу англійського алфавіту; 2) перевірки ідентичності символів у комбінації.

10.192. Написати програму для генерування серійних номерів програмних продуктів. Кожен номер складається з 2 літер англійського алфавіту, що позначають голосні звуки, і чотирицифрового числа. Наприклад: AU-2798. Написати програму, яка заповнює масив 12 значеннями серійних номерів. Для створення серійного номера розробити відповідну функцію.

10.193. Написати програму для порівняння двох кортежів довільної довжини. Для створення кортежу використати функцію, яка запитує у користувача рядок і перетворює його на кортеж.

10.194. Написати програму, яка об'єднує два одновимірних масиви, наповнених випадковими числами. Для створення і наповнення масиву написати функцію, яка отримує один аргумент – довжину масиву. Використати бібліотеку `array`.

10.195. Написати програму, яка створює і заповнює кортеж випадковими символами. Для генерування випадкових символів (літери англійського алфавіту і цифри від 0 до 9) створити відповідну функцію.

10.196. Написати програму, яка заповнюватиме кортеж з 10 елементів випадковими літерами англійського алфавіту, що позначають голосні звуки. Для створення і заповнення кортежу розробити відповідну функцію.

10.197. Написати програму для пошуку елемента у двовимірному масиві. Для створення і наповнення двовимірного масиву розмірністю $[M][N]$ розробити відповідну функцію, яка приймає три аргументи: M , N , \max (\max – межа діапазону випадкових чисел $[0; \max]$). У програмі використати бібліотеку `numpy`.

10.198. Користувач задає символ англійського алфавіту. Програма повинна заповнити рядок випадковими символами англійського алфавіту у діапазоні від a до заданого користувачем. Розробити 2 функції: 1) генерування випадкового символу від a до RS (символ RS задається з клавіатури); 2) заповнення рядка заданої довжини випадковими символами.

10.199. Алфавіт повідомлення являє перші 10 символів англійського алфавіту і записаний у кортеж. Написати програму, яка генеруватиме повідомлення з цих символів, довжину повідомлення задає користувач. Для цього розробити функцію, яка генеруватиме випадковий символ з кортежу.

10.200. Алфавіт повідомлення являє собою 15 символів і записаний у кортеж. Програма повинна згенерувати повідомлення довжиною N з символів, записаний у кортеж. Для генерування випадкового символу з кортежа розробити відповідну функцію. Число N задається з клавіатури.

Розділ 11. Модулі

На мові Python написано значну кількість модулів, за допомогою яких у програмі можна використати додаткові функції, яких немає в основній бібліотеці Python. Для того, щоб підключити модуль, слід на початку програми записати ключове слово `import` і набрати назву модуля:

```
import random
```

Для того, щоб потім використати цей модуль у програмі, слід посылатися на нього відповідним чином:

```
x=random.randint()
```

Часто виникає необхідність використати модуль стороннього розробника. Цей модуль може встановлюватися як програмна бібліотека (наприклад, NumPy), а може бути реалізована у вигляді файла з розширенням `.py`. Відповідний файл потрібно викачати і зберегти у папку з програмою, яка буде виконуватися.

Наприклад, бібліотеку для графіки реалізовано у вигляді файла `graphics.py`. Для того, щоб використати функції і методи з цієї бібліотеки, слід викачати файл `graphics.py` і скопіювати в папку `Examples` (у ній зберігаються всі файли програм, написані в середовищі Thonny).

Далі слід імпортувати модуль. Це можна зробити двома командами. Перша команда:

```
import graphics
```

Друга команда:

```
import graphics as gr
```

Після цього можна використовувати функції і методи, які є в модулі.

Мова Python надає широкі можливості для створення і поширення власних модулів. Як ми вже бачили раніше, підключити модуль досить просто, тому розробка власних модулів і їх поширення не викликають особливих труднощів (якщо ви володієте мовою програмування Python).

Створення власних модулів

Створення модулів на мові Python подібне до створення звичайних програм. Створюється звичайний файл з розширенням `py`, у який записуються всі функції (методи), які повинні бути в модулі. Для прикладу опишемо функції:

```
import math
def volume_ball(R):
    V=math.pi*math.pow(R,3)*4/3
    return V
def volume_kub(a):
```

```
V=math.pow(a,3)
return(V)
```

Цей файл необхідно зберегти у папку з програмами, ввівши розширення файла .py: my_module1.py.

Розглянемо, як можна використати методи з цього модуля в іншій програмі. Текст програми:

```
import my_lib1
a=float(input("Input a"))
R=float(input("Input R"))
V1=my_lib1.volume_ball(R)
V2=my_lib1.volume_kub(a)
print("ball=",V1)
print("kub=",V2)
```

Для того, щоб використати модуль у програмі, слід скопіювати його в папку з програмою і імпортувати його за допомогою команди import.

```
import my_lib1
```

Або

```
from my_lib1 import *
```

Для того, щоб викликати модуль, необхідно вказати ім'я модуля і через крапку – ім'я функції, яка викликається:

```
z=my_module.symma(a,b)
```

Завдання для виконання

11.1. Створити модуль triangle_lib, у якому записати функції для обчислення площі трикутників для таких випадків: 1) трикутник задано двома сторонами і кутом між ними; 2) трикутник задано трьома сторонами. Написати програму, в якій підключити цей модуль і використати описані функції для обчислення площі трикутників.

11.2. Створити модуль rect_lib, у якому записати функції для обчислення площі прямокутників для таких випадків: 1) прямокутник задано двома сторонами; 2) прямокутник задано однією стороною (тобто, це квадрат). У програмі використати функції, імпортовані з модуля.

11.3. Розробити модуль paral_lib, в який записати функції для обчислення площі паралелограма, заданого: 1) двома сторонами і кутом між ними; 2) двома діагоналями і кутом між ними. У програмі використати імпортовані функції для обчислення площі паралелограмів для кількох випадків.

11.4. Створити модуль romb_lib, в якому записати функції для обчислення площі ромба для двох випадків: 1) коли ромб задано двома сторонами і кутом між ними; 2) коли ромб задано двома

діагоналями. У програмі використати імпортовані функції для обчислення площі ромба.

11.5. Розробити модуль `rectangle_lib`, який повинен містити функції для обчислення периметра, площі і довжини діагоналі прямокутника. Підключити модуль до програми і вивести на екран площі і периметри 8 прямокутників (розміри прямокутників записуються у двовимірний масив), які є випадковими числами у діапазоні [1;6].

11.6. Написати модуль `triangle_lib`, який містить функції для обчислення гіпотенузи, площі і периметра прямокутного трикутника. Аргументи – довжини катетів прямокутника. Підключити модуль до основної програми і обчислити довжину гіпотенузи, величину площі і периметр прямокутного трикутника за розмірами, введеними користувачем.

11.7. Написати модуль `kolo_lib`, який містить 4 функції: для обчислення площі і периметра кола, заданого радіусом і діаметром. В основній програмі передбачити меню з 4 пунктів, у залежності від вибору повинна запускатися відповідна функція з модуля.

11.8. Написати модуль `pp_lib`, який містить такі функції: 1) для обчислення об'єму прямокутного паралелепіпеда; 2) для обчислення площі поверхні прямокутного паралелепіпеда; 3) для обчислення діагоналі прямокутного паралелепіпеда. Для всіх трьох функцій аргументи – лінійні розміри паралелеграма a, b, c . Написати програму, у якій необхідно забезпечити виконання цих функцій у залежності від вибору користувача.

11.9. Написати модуль `k_lib`, який буде містити такі функції: 1) для обчислення об'єму кулі; 2) для обчислення площі поверхні кулі; 3) для обчислення діаметра кулі. Аргумент для всіх трьох функцій один – величина радіуса кулі. У програмі передбачити вибір користувачем потрібної функції.

11.10. Написати програму, яка буде використовувати модуль. Створити модуль `f_lib` і підключити його до програми. Модуль повинен містити функції: 1) обчислення середнього арифметичного двох чисел; 2) середнього геометричного двох чисел. Аргументи для обох функцій – два цілих числа a, b . У програмі забезпечити можливість вибору користувачем потрібної функції.

11.11. Написати програму, що використовує зовнішній модуль. Створити відповідний модуль, що міститиме такі функції: 1) визначення більшого з трьох чисел; 2) визначення меншого з трьох чисел. Аргументи для обох функцій – три цілих числа a, b, c . У програмі створити меню для вибору однієї з цих функцій користувачем.

11.12. Розробити програму, яка використовує зовнішній модуль. Написати відповідний модуль, який повинен містити такі функції: 1) визначення більшого з чотирьох чисел; 2) визначення меншого з чотирьох чисел. Аргументи для обох функцій – 4 числа, над якими виконуються дії. В основній програмі створити меню вибору відповідної функції.

11.13. Написати програму, яка обчислює суму двох випадкових чисел у діапазоні $[a;b]$. Для цього створити відповідний модуль, який міститиме одну функцію (для обчислення суми двох випадкових чисел). Вхідні аргументи для функції: числа a, b – межі діапазону.

11.14. Написати програму, яка за вибором користувача обчислює площу бічної поверхні і об'єм циліндра. Для цього створити і підключити до програми модуль, який містить дві функції: 1) для обчислення площі бічної поверхні циліндра; 2) для обчислення об'єму циліндра. Вхідні параметри для обох функцій – радіус циліндра R і висота циліндра h .

11.15. Написати модуль, який міститиме функції для визначення: 1) найменшого елемента масиву; 2) найбільшого елемента масиву. Вхідні аргументи для функцій: 1) кількість елементів масиву N ; 2) сам масив цілих чисел. У програмі створити меню, яке надаватиме можливість користувачу вибрати потрібну дію, і підключити модуль до основної програми. Масив вводиться з клавіатури.

11.16. Написати програму, яка буде обчислювати суму парних або непарних елементів масиву. Для цього створити і підключити до програми модуль, який міститиме дві функції: 1) обчислює суму парних елементів масиву; 2) обчислює суму непарних елементів масиву. Масив з 10 елементів заповнюється випадковими цілими числами.

11.17. Розробити модуль, який містить функції: 1) для обчислення площі трикутника загального виду; 2) для обчислення площі прямокутного трикутника. Вхідні параметри для функцій: 1) довжини двох сторін і величина кута між ними в градусах; 2) довжини двох катетів. Підключити модуль до програми і створити меню вибору відповідної функції.

11.18. Створити модуль, який містить функції для обчислення: 1) кінетичної енергії тіла; 2) потенціальної енергії тіла. Вхідних параметрів немає, функції самі запитують у користувача потрібні дані. Підключити модуль до програми і створити меню для вибору користувачем відповідної функції.

11.19. Написати програму, яка обчислює попарно суми сусідніх елементів масиву. Для цього створити модуль, який

міститиме 2 функції: 1) заповнює масив випадковими числами; 2) обчислює попарно суми сусідніх елементів масиву. Підключити модуль до основної програми, запрограмувати послідовне виконання цих двох функцій.

11.20. Розробити модуль, який міститиме 2 функції: 1) заповнює масив з 7 елементів випадковими числами у діапазоні [-20;20]; 2) підраховує кількість додатніх елементів у цьому масиві. Підключити модуль до основної програми, записати команди так, щоб виклик функцій здійснювався послідовно.

11.21. Створити модуль, який міститиме 3 функції: 1) для обчислення суми трьох чисел; 2) для обчислення добутку трьох чисел. Аргументи функцій – три цілих числа. Підключити цей модуль до програми, створити меню, де користувач може вибрати виконувану дію.

11.22. Написати програму, яка містить модуль з функціями: 1) для обчислення кількості пробілів у рядку; 2) для обчислення кількості літер «i» у рядку. Аргумент функції: рядок символів. Підключити модуль до програми і створити меню, в якому користувач може вибрати виконувану дію.

11.23. Написати програму, яка містить модуль з функціями: 1) для обчислення суми цифр цілого числа; 2) для обчислення добутку цифр цілого числа. Вхідний параметр для обох функцій: ціле число. Підключити модуль до програми і створити меню для вибору користувачем потрібної дії.

11.24. Розробити модуль, який міститиме 2 функції: 1) для визначення першої цифри трицифрового числа; 2) для визначення останньої цифри трицифрового числа. Вхідний параметр: трицифрове число. Підключити модуль до програми і послідовно викликати обидві функції.

11.25. Створити програму, яка міститиме модуль, що складається з функцій: 1) для обчислення суми двох випадкових чисел; 2) для обчислення добутку двох випадкових чисел. Підключити модуль до програми і викликати обидві функції послідовно.

11.26. Написати програму і підключити до неї модуль, який містить такі функції: 1) для створення масиву і заповнення його випадковими числами; 2) для виведення на екран всього масиву; 3) для обчислення середнього арифметичного елементів масиву. У програмі викликати ці функції послідовно.

11.27. Створити програму і підключити до неї модуль, що містить функції: 1) для створення масиву заданої довжини і заповнення його випадковими числами; 2) для підрахунку додатніх елементів масиву; 3) для підрахунку від'ємних елементів масиву. У

програмі створити меню для вибору потрібної функції користувачем.

11.28. Написати модуль, який міститиме функції: 1) для стирання перших N символів рядка; 2) для стирання останніх N символів рядка. Підключити модуль до програми і послідовно викликати ці функції. Початковий рядок символів вводиться користувачем.

11.29. Створити модуль з такими функціями: 1) для обчислення об'єму прямокутного паралелепіпеда; 2) для обчислення об'єму циліндра. Підключити модуль до програми і створити меню для вибору цих функцій користувачем.

11.30. Написати модуль, який міститиме такі функції: 1) для обчислення периметра прямокутника; 2) для обчислення периметра кола. Підключити модуль до програми і створити меню користувача, за допомогою якого можна вибрати потрібну функцію.

11.31. Створити модуль, що містить функції: 1) для обчислення площі поверхні куба; 2) для обчислення площі поверхні прямокутного паралелепіпеда. Підключити модуль до програми і створити меню, за допомогою якого користувач може викликати потрібну функцію.

11.32. Написати програму, яка буде на екрані рівносторонній трикутник і квадрат. Для цього створити і підключити до програми модуль з двома підпрограмами: 1) для побудови рівностороннього трикутника (вхідні параметри: координати стартової точки x і y , довжина сторони a); 2) для побудови квадрата (вхідні параметри такі самі). У програмі використати модуль `turtle`.

11.33. Написати програму, яка малює на екрані ромб. Для цього створити модуль, який містить процедуру для малювання ромба. Вхідні дані для процедури: координати початкової точки x і y , довжина сторони ромба a , кут між сторонами ромба α . У програмі використати модуль `turtle`.

11.34. Написати програму, яка малює на екрані рівносторонній трикутник зі стороною a і квадрат зі стороною b . Для цього створити окремий модуль, що міститиме дві підпрограми – перша малюватиме квадрат, друга – рівносторонній трикутник. У програмі використати також модуль `turtle`.

11.35. Створити модуль, який буде містити дві функції: 1) для поділу рядка на окремі частини довжиною 4 символи кожна; 2) для склеювання рядків в один. У програмі створити меню, за допомогою якого користувач зможе викликати потрібну підпрограму.

11.36. Написати програму, яка обчислюватиме: 1) суму цифр натурального числа; 2) добуток цифр натурального числа. Для цього створити модуль, у в ньому – дві функції.

11.37. Один із методів шифрування полягає у заміні чисел початкового повідомлення. Числа, записані у масив, розбиваються на біграми (тобто сполучення по 2 числа) і записуються в двовимірний масив, який містить 2 стовпчики. Після цього рядки з парними і непарними номерами міняються місцями. Реалізувати цей алгоритм на практиці, створивши модуль з двома функціями, які забезпечують відповідно: 1) шифрування даних; 2) виведення зашифрованого повідомлення на екран.

11.38. Написати програму, яка обчислюватиме площі геометричних фігур: квадрата, прямокутника, трикутника, трапеції. Для цього створити окремий модуль, а в ньому – відповідні функції. Вхідні параметри для функцій визначити самостійно.

11.39. Номери мобільних телефонів одного з операторів являють собою восьмицифрові числа (у даній задачі – випадкові числа у діапазоні від 10000000 до 99999999). Масив містить 12 таких чисел. Програма повинна відсортувати масив у порядку збільшення номерів і вивести масив на екран у стовпчик. Для цього створити модуль, який міститиме дві функції: 1) заповнення масиву; 2) сортування масиву.

11.40. Масив містить 10 рядкових величин – номери автомобілів, що складаються з 2 частин: перша частина – випадкове чотиризначне число, друга частина – серія (наприклад, СІА). Написати програму для обробки масиву автомобільних номерів. Для цього створити модуль, який міститиме 2 функції: 1) для заповнення масиву номерів; 2) для сортування масиву номерів; 3) для виведення масиву номерів на екран у стовпчик.

11.41. Написати програму, яка генерує масив з 12 телефонних номерів, кожен з яких має формат «097+xxxxxxx», де xxxxxx – випадкове семизначне число. Для цього створити модуль, який міститиме функції: 1) заповнення масиву номерами; 2) сортування масиву; 3) виведення масиву на екран. Під час створення масиву використати бібліотеку `numpy`.

11.42. Користувач вводить кількість елементів масиву. Програма повинна заповнити масив рядками по 4 символи, кожен з яких являє собою випадковий символ англійського алфавіту. Для цього створити модуль, який міститиме функції: 1) генерування випадкового символу з англійського алфавіту і заповнення рядка 4 різними символами; 2) заповнення масиву рядками; 3) виведення масиву на екран у стовпчик. Під час створення масиву використати бібліотеку `numpy`.

11.43. Написати програму, яка заповнює рядок 44 випадковими символами англійського алфавіту. Для цього створити

модуль, який міститиме функції: 1) генерування випадкового символу; 2) заповнення рядка випадковими символами.

11.44. Написати програму, яка заповнюватиме рядок довжиною 60 випадковими символами українського алфавіту. Для цього написати і підключити до програми модуль, який міститиме 2 функції: 1) генерування випадкового символу; 2) заповнення рядка випадковими символами. У програмі створити меню, за допомогою якого користувач може вибрати потрібну функцію.

11.45. Написати програму, яка шифруватиме повідомлення таким чином: береться номер символу за алфавітом, збільшується на x , потім виводиться отриманий символ. Для цього розробити модуль, який включатиме функції: 1) шифрування повідомлення; 2) дешифрування повідомлення.

11.46. Написати програму, яка генерує рядок випадкових символів за таким принципом: на першому місці записується випадкова літера англійського алфавіту, на другому – випадкова цифра. Для цього створити модуль і підключити його до програми. Модуль повинен містити 2 функції: 1) генерування випадкового символу англійського алфавіту; 2) створення рядка заданої довжини з символів, згенерованих першою функцією, і випадкових цифр.

11.47. Один з варіантів шифрування символічних повідомлень полягає у тому, що повідомлення розбивається на біграми (буквосполучення з 2 символів), потім біграми з непарними номерами і біграми з парними номерами міняються місцями. Реалізувати цей алгоритм за допомогою двовимірного масиву рядків (кожен рядок довжиною 2 символи). Написати модуль і підключити його до програми. Модуль повинен мати функції: 1) розбиття рядка на біграми і занесення їх у двовимірний масив; 2) заміна місцями біграм з непарними і парними номерами; 3) перетворення масиву на рядок (це й буде зашифрований текст). У програмі використати бібліотеку `numpy`.

11.48. Інший метод шифрування такий: вихідне повідомлення (рядок) розбивається на біграми (буквосполучення з двох символів), біграми записуються у зворотному порядку, між ними вставляються біграми з випадкових символів, після чого біграми об'єднуються в один рядок і виходить зашифрований текст. Реалізувати цей алгоритм у програмі. Для цього створити і підключити модуль з функціями: 1) розбиття тексту на біграми і перетворення у двовимірний масив; 2) створення біграм з випадкових символів; 3) запис біграм з випадкових символів у масив; 4) перетворення масиву на рядок.

11.49. Написати програму, яка створює рядок, який складається з 30 випадкових символів англійського і українського

алфавіту, які чергуються. Для цього створити і підключити до програми модуль, який міститиме дві функції: 1) генерує випадковий символ українського алфавіту; 2) генерує випадковий символ англійського алфавіту.

11.50. Написати програму, яка генерує і виводить на екран N повідомлень (рядків), кожне з яких складається з двох випадкових символів англійського алфавіту і двох цифр. Для цього створити і підключити до програми модуль з такими функціями: 1) генерує випадковий символ англійського алфавіту; 2) створює рядок з двох випадкових символів (згенерованих першою функцією) і дописує до них дві випадкові цифри.

Розділ 12. Файли

Короткі теоретичні відомості

Програми на мові Python можуть застосовувати файли для зберігання даних. Для відкриття файла використовується функція `open`:

```
f=open(x,"rt")
```

Функція `open` має багато параметрів: ім'я файла, режим роботи з файлом.

Таблиця 12.1

Режими роботи з файлами

Режим	Характеристика
'r'	Відкриття на читання
'w'	Відкриття на запис (і перезапис існуючого файла)
'x'	Відкриття файла на запис, якщо файла не існує, інакше генерується виняткова ситуація
'a'	Відкриття файла і запис даних у кінець файла
'b'	Відкриття файла у двійковому режимі
't'	Відкриття файла у текстовому режимі
'+'	Відкриття файла на читання і на запис

Режими можуть бути об'єднані.

Третій параметр – `encoding` (кодування) – потрібен лише в текстовому режимі читання файла.

Читання з файла

Існує кілька способів зчитування даних з файла, найбільш поширеними є два:

- `read()` – читання файла повністю. Якщо в дужках вказати число `N`, то зчитує перші `N` символів;
- `for line in f:` – зчитування даних по рядках.

Приклад читання файла повністю

```
x='text.txt'  
f=open(x,"rt")  
z=f.read()  
print(z)  
f.close()
```

Приклад зчитування даних по рядках:

```
x='text.txt'  
f=open(x,"rt")  
for line in f:  
    print(line)  
f.close()
```

Запис у файл

Для запису даних у файл його слід відкрити на запис:

```
f = open('text.txt', 'w')
```

Для запису рядка у файл застосовується метод write():

```
for index in l:  
... f.write(index + '\n')
```

Метод write() повертає кількість записаних символів.

Після запису у файл його слід закрити:

```
f.close()
```

Приклад запису у файл:

```
x='text1.txt'  
f=open(x,"wt")  
for i in range(10):  
    z=str(i)+"\n"  
    print(z)  
    f.write(z)  
f.close()
```

Зчитасмо дані з цього файла:

```
x='text1.txt'  
f=open(x,"rt")  
z=f.read()  
print(z)  
f.close()
```

Для зберігання складних типів даних в Python доцільно використати модулі pickle і json.

Читання з файла

12.1. Написати програму, яка малює на екрані квадрат заданих розмірів. Розміри квадрата зчитати з файла input.txt. Використати модуль turtle.

12.2. Написати програму, яка обчислює суму і різницю двох чисел, зчитаних з файла.

12.3. У файл input.txt записано значення сторін прямокутника. Програма повинна обчислити його площу і периметр.

12.4. Файл input.txt містить два числа – межі діапазону [a;b]. Програма повинна обчислити суму всіх парних чисел з цього діапазону.

12.5. Файл set.txt містить елементи масиву – 12 цілих чисел. Програма повинна обчислити їх суму.

12.6. У файл input.txt у стовпчик записано елементи масиву, кількість яких наперед невідома. Написати програму, яка виведе на екран найбільший і найменший елементи масиву.

12.7. Файл input.txt містить розміри прямокутника a,b. Програма повинна намалювати на екрані цей прямокутник. Використати модуль turtle.

12.8. У файл записано значення кольору ліній. Програма повинна намалювати на екрані коло радіусом R заданого кольору. Число R задається користувачем. У програмі використати модуль turtle.

12.9. У файл input.txt записано список чисел невідомої наперед довжини (у рядок). Написати програму, яка обчислить різницю найбільшого і найменшого елементів списку.

12.10. Файл setup.txt містить рядок цілих чисел. Написати програму, яка виведе ці числа на екран у стовпчик, попередньо дописавши до набору чисел три десятки.

12.11. Написати програму, яка зчитує з файла 7 чисел, записаних у стовпчик, і обчислює їх добуток. При цьому програма повинна перевіряти, щоб модулі чисел не перевищували 50.

12.12. Написати програму, яка обчислює найменше з трьох чисел, зчитаних з файла input.txt.

12.13. Написати програму, яка обчислює середнє геометричне трьох чисел, зчитаних з файла start.txt. Числа записані у стовпчик.

12.14. У файл income.txt у стовпчик записано 8 цілих чисел. Написати програму, яка виведе ці числа на екран у порядку зростання.

12.15. Файл start.txt містить набір цілих чисел, записаних у рядок. Написати програму, яка виводить на екран число, що знаходиться посередині набору. Якщо таких числа два, то повинні бути виведені два.

12.16. У файл input.txt записано три рядки – значення кольорів. Програма повинна зчитати ці кольори з файла і намалювати на екрані три вертикальні лінії відповідних кольорів. У програмі використати модуль turtle.

12.17. У файлі input.txt записані 3 рядки двовимірного масиву. Написати програму, яка зчитає цей масив з файла і обчислить суми чисел кожного рядка.

12.18. Файл input.txt містить одновимірний масив з цілих чисел, записаний у рядок. Написати програму, яка обчислить найбільше і найменше значення цього масиву.

12.19. Файл setup.txt містить дані про встановлення програмного забезпечення (в окремих рядках – назва програми; шлях до файла; обсяг у байтах). Написати програму, яка виведе на екран ці дані в одному рядку.

12.20. Файл res.txt містить дані про студента: прізвище та ініціали; номер групи; номер курсу – записані в різних рядках. Програма повинна вивести ці дані на екран в трьох різних рядках.

12.21. У файл записано три цілих числа. Програма повинна піднести до квадрата середнє число.

12.22. У файлі `income.txt` записано у рядок 3 числа. Програма повинна вивести на екран суму останніх цифр цих чисел.

12.23. У файл `input.txt` записано рядок слів, які необхідно зашифрувати. Програма повинна зашифрувати рядок шифром Цезаря (тобто, зсунути всі символи вправо за алфавітом).

12.24. Файл `proba.txt` містить масив з 10 чисел, записаний у стовпчик. Програма повинна вивести на екран найбільше і найменше значення елементів масиву.

12.25. У файл `extern.txt` записано масив з 10 слів – назви міст України, кожне слово на новому рядку. Програма повинна обчислити довжини слів і вивести їх на екран.

12.26. Файл `round.txt` містить рядок дійсних чисел – діаметри кіл. Програма повинна обчислити площі цих кіл.

12.27. Масив з 7 цілих чисел записано у файл `input.txt` (у рядок). Програма повинна обчислити добуток всіх парних елементів масиву.

12.28. Масив з 8 чисел записано у файл `income.txt` (у стовпчик). Програма повинна замінити у цьому масиві непарні числа на випадкові парні.

12.29. Масив з 9 дійсних чисел записано в файл `file1.txt` (у стовпчик). Програма повинна відсортувати масив за зростанням і вивести на екран у рядок.

12.30. Масив з 11 цілих чисел записано у стовпчик у файлі `start.txt`. Програма повинна обчислити суму елементів масиву з парними індексами і вивести її на екран.

12.31. Три рядка тексту записано у файл `start.txt`. Програма повинна зчитати ці рядки з файла і вивести на екран у зворотному порядку.

12.32. Файл `start.txt` містить масив 3×3 , записаний у 3 рядки. Програма повинна вивести на екран елементи масиву, які знаходяться на головній діагоналі таблиці.

12.33. Файл `input.txt` містить масив 4×4 , записаний у 4 рядки. Програма повинна вивести на екран елементи масиву, які знаходяться на допоміжній діагоналі таблиці.

12.34. Файл `file1.txt` містить масив з 8 цілих чисел, записаних у стовпчик. Програма повинна вивести масив у рядок у зворотному порядку.

12.35. Файл `file777.txt` містить масив цілих чисел, записаних у рядок. Написати програму, яка: 1) виведе масив у рядок на екран у початковому вигляді; 2) виведе масив на екран у рядок, відсортувавши його за спаданням.

12.36. Файл file12.txt містить масив цілих чисел, записаний у рядок. Програма повинна вивести на екран два перших і два останніх елементи масиву.

12.37. У файл file9.txt записано у стовпчик масив. Програма повинна зчитати з файла масив і вивести елемент, який стоїть посередині масиву (якщо масив містить непарну кількість елементів) або два елементи (якщо масив містить парну кількість елементів).

12.38. Файл start.txt містить цілі числа a і b . Написати програму, яка заповнить масив з 5 цілих чисел значеннями у діапазоні $[a;b]$.

12.39. У файл z.txt записано ціле число z . Програма повинна заповнити масив з 9 чисел випадковими значеннями у діапазоні $[-z;z]$, вивести масив на екран і підрахувати кількість додатних елементів у ньому.

12.40. Файл x.txt містить три цілих числа – a , b , y , записані у рядок. Програма повинна заповнити масив з 11 елементів випадковими числами у діапазоні $[a;b]$, вивести масив на екран у рядок і обчислити кількість елементів, значення яких більше за y .

12.41. Файл proba.txt містить цілі числа a_1, a_2 і k , записані у стовпчик. Програма повинна заповнити масив випадковими цілими числами у діапазоні від a_1 до a_2 , вивести масив на екран, потім помножити всі елементи масиву на число k і вивести масив на екран.

12.42. Елементи цілочисельного масиву $A[]$ записано у файл (у рядок). Написати програму, яка виведе на екран усі елементи масиву, які діляться на 4 націло.

12.43. У файл setup.txt записано цілі числа a , b , n . Написати програму, яка заповнить масив довжиною n випадковими цілими числами у діапазоні $[a;b]$.

12.44. Файл question.txt містить чотири цілих числа, записані у стовпчик. Програма повинна вивести на екран найменше і найбільше числа.

12.45. Файл bug.txt містить дані про збої у роботі комп'ютера (коди помилок у вигляді чотирицифрових чисел). Написати програму, яка визначить, скільки разів зустрічається помилка, яка записана першою.

12.46. Елементи списку $Z[]$ записані у файл input.txt у рядок. Програма повинна: 1) відсортувати список за зростанням; 2) вивести найбільший і найменший елементи списку.

12.47. Існує порожній список $A[]$. Написати програму, яка заповнює його цілими числами, зчитаними з файла input.txt, і виводить на екран у рядок.

12.48. У файл start.txt записано ціле число x . Написати програму, яка заповнить масив з n чисел випадковими числами у діапазоні $[-x;x]$ і виведе його на екран у рядок.

12.49. Файл input.txt містить координати початкової точки $(x;y)$ і розміри квадрата a . Програма повинна намалювати на екрані квадрат зі стороною a , починаючи малювання з точки $M(x;y)$.

12.50. У файл z.txt записано координати точки $M(x;y)$. Програма повинна обчислити відстань від цієї точки до початку координат.

Запис у файл

12.51. Написати програму, яка визначає корені квадратного рівняння. Коефіцієнти a, b, c вводить користувач, результат повинен бути записаний у файл output.txt.

12.52. Написати програму, яка визначає середнє арифметичне двох чисел, зчитаних з клавіатури, і записує результат у файл solution.txt.

12.53. Написати програму, яка обчислює суму всіх натуральних чисел в інтервалі від a до b , які закінчуються на 1 і діляться на 9.

12.54. Написати програму, яка заповнює масив з 11 елементів випадковими цілими числами у діапазоні $[-20;40]$ і записує масив у рядок у файл output.txt.

12.55. Написати програму, яка заповнює масив з N елементів випадковими цілими числами у діапазоні $[a;b]$ і записує його у файл result.txt у стовпчик. Числа a, b, N задаються з клавіатури.

12.56. У результаті змагань отримано дальність польоту моделей літаків у метрах – 12 цілих чисел (їх вводить користувач). Написати програму, яка запише отримані значення в один рядок у файл fly.txt.

12.57. Написати програму, яка створює словник з такими даними: прізвище учня, рік народження, заносить у словник дані, введені користувачем, і записує дані у файл database.txt.

12.58. Написати програму, яка містить словник з даними: ідентифікатор – марка автомобіля – вантажопідйомність. Дані повинні вводитися у словник користувачем. Після завершення введення даних словник повинен зберігатися у файл output.txt.

12.59. Написати програму, в якій створюється список A , заповнюється цілими числами з клавіатури і записується в один рядок у файл solution.txt.

12.60. Написати програму, яка зчитує з екрану рядки тексту і записує їх у файл output.txt.

12.61. Написати програму, яка зчитує рядки тексту з екрану і записує у файл рядки, які мають парний номер.

12.62. Написати програму, яка шифрує повідомлення, введене користувачем (рядок тексту) за таким принципом: після кожного символу початкового тексту вставляються два випадкові символи англійського алфавіту. Результат повинен записуватися у рядок solution.txt.

12.63. Користувач вводить два цілих числа. Написати програму, яка обчислює їх суму, різницю, добуток, частку і остачу від ділення, після чого записує їх у файл output.txt.

12.64. Розробити програму, яка заповнює випадковими числами два масиви: перший – з 7 елементів, другий – з 10 елементів. Діапазон випадкових чисел – від -10 до 10. Результат повинен бути записаний у файл finish.txt.

12.65. Користувач вводить рядок символів. Написати програму, яка видаляє з нього всі пробіли і записує рядок у файл output.txt.

12.66. Написати програму, яка запише у файл output.txt всі тризначні числа від x_1 до x_2 , які закінчуються на цифру 6 і діляться на 4.

12.67. Користувач вводить розміри прямокутного паралелепіпеда a , b , c . Програма повинна обчислити об'єм прямокутного паралелепіпеда і записати його у файл output.txt.

12.68. Користувач задає межі діапазону x_1 і x_2 – два цілих числа. Радіус кола являє собою випадкове число у діапазоні $[x_1; x_2]$. Написати програму, яка обчислить площу круга і запише її у файл solution.txt.

12.69. Користувач задає ціле число – межу діапазону a_1 . Програма повинна згенерувати 4 числа, кожне з яких є випадковим числом у діапазоні $[0; a_n - 1]$. Наприклад, число a_2 є випадковим у діапазоні від 0 до a_1 . Всі утворені числа повинні бути записані у файл digit.txt (у рядок).

12.70. Користувач вводить чисельник дробу – ціле двозначне число. Програма повинна записати знаменники дробів такі, щоб утворені дроби були правильними, у файл output.txt (у стовпчик).

12.71. Користувач вводить знаменник дробу – ціле двозначне число. Програма повинна згенерувати всі цілі числа – чисельники дробу такі, щоб утворений дріб був правильним, і записати отримані значення у файл output.txt (у рядок).

12.72. Написати програму, яка генерує випадкове число на проміжку $[z-100; z+100]$ і записує його у файл solution.txt. Число z задає користувач.

12.73. З клавіатури вводяться 10 натуральних чисел – елементів масиву. Програма повинна згенерувати масив випадкових чисел, лівою межею діапазону є 0, а правою – відповідне число з першого масиву. Утворений масив програма повинна записати у файл result.txt.

12.74. Користувач вводить рядок символів. Програма продовжує рядок до довжини, кратної 4, після чого розбиває рядок на буквосполучення по 4 символи у кожному і записує кожне буквосполучення окремими рядками в файл output.txt.

12.75. Користувач вводить рядок символів. Програма повинна продовжити рядок до довжини, кратної 3, після чого розбити рядок на буквосполучення по 3 символи у кожному і записує кожне буквосполучення в окремий рядок у файл output.txt.

12.76. Написати програму, яка генерує три рядки по 12 символів, які складаються з випадкових цифр і записує їх у файл output.txt.

12.77. Користувач вводить два натуральних числа. Програма повинна обчислити суму кубів цих чисел і записати її у файл word.txt.

12.78. З клавіатури вводяться два цілих числа M і N – розміри двовимірного масиву. Програма повинна заповнити масив випадковими числами у діапазоні від 0 до 100 і записати масив у файл output.txt.

12.79. Користувач вводить рядок з X символів. Програма повинна записати у файл guadok.txt перші p ’ять і три останні символи рядка.

12.80. Користувач вводить рядок. Програма повинна записати у файл z.txt другу половину рядка.

12.81. Користувач вводить рядок слів, відділених пробілами. Програма повинна записати у файл slova.txt перше і останнє слова.

12.82. Користувач вводить рядок слів, розділених пробілами. Програма повинна записати у файл result.txt у стовпчик довжини цих слів.

12.83. Користувач вводить рядок слів, розділених пробілами. Програма повинна записати у файл result.txt у стовпчик найдовше слово або слова.

12.84. Користувач вводить з клавіатури чотири рядка, які складаються з слів, розділених пробілами. Програма повинна записати у файл найдовше і найкоротше слово.

12.85. Написати програму, яка записує у файл output.txt N рядків тексту, який складається з випадкових символів. Число N задається користувачем.

12.86. Написати програму, яка записує у файл `finish.txt` N рядків по M символів у кожному. Символ – випадкове одноцифрове число.

12.87. Користувач задає межі діапазону. Програма повинна заповнити масив 4×4 випадковими числами з цього діапазону і записати у файл `solution.txt` суми стовпчиків масиву.

12.88. Написати програму, яка заповнює масив $Z[N][N]$ випадковими числами у діапазоні $[0;30]$ і записує його у файл `output.txt`. Розмірність масиву N задається користувачем.

12.89. Написати програму, яка містить функцію і процедуру. Функція повинна заповнювати масив з N елементів випадковими числами, а процедура повинна записати масив у файл `output.txt` у рядок.

12.90. Написати програму, яка запитує у користувача день, місяць і рік його народження і записує тримані дані у файл `file12.txt`.

12.91. Користувач вводить день, місяць і рік свого народження, прізвище та ініціали. Програма повинна обчислити вік користувача і записати у файл `age.txt` прізвище, ініціали і вік користувача.

12.92. Написати програму, яка обчислюватиме випадкове число z у діапазоні від 20 до 50, створюватиме масив $M [3][3]$, заповнить цей масив випадковими числами у діапазоні $[0;z]$ і запише масив у файл `masiv.txt`.

12.93. Масив $T[20]$ містить маси плодів – випадкові числа у діапазоні від 200 до 450 (г). Програма повинна обчислити загальну масу всіх плодів, потім записати у файл `fruits.txt` обчислене значення.

12.94. Масив $Z[12]$ містить ціни різних зошитів (випадкові числа від 50 коп до 8,5 грн). Програма повинна обчислити середню ціну зошита і записати її у файл `copybooks.txt`.

12.95. Користувач задає ціле число z – межу діапазону. Програма повинна заповнити масив випадковими числами у діапазоні $[-z;z]$ і записати його у файл `array.txt`.

12.96. Масив $A[4][4]$ заповнено випадковими цілими числами у межах від 2 до 20. Написати програму, яка заповнить масив $B[4][4]$ випадковими числами у діапазоні $[0;z]$ ($z = A[i][j]$) і запише створений масив у файл `random.txt`.

12.97. Користувач вводить рядок тексту і задає число k (довжина елемента масиву символів). Програма повинна розбити початковий рядок на масив рядків (кожен довжиною k), після чого зберегти масив у файл `strings.txt` – кожен елемент масиву в окремому рядку. Для створення масиву використати бібліотеку `numpy`.

12.98. Написати програму, яка заповнює масив розмірністю 3×3 випадковими символами англійського алфавіту і записує цей

масив у файл symbols.txt. Для створення масиву використати бібліотеку numpy.

12.99. Користувач вводить рядок тексту довільної довжини. Програма повинна змінити порядок символів у рядку на зворотній і записати рядок у файл гуadok.txt – кожен символ в окремому рядку.

12.100. Користувач задає довжину рядка. Програма повинна заповнити рядок випадковими символами англійського алфавіту і записати його у файл my_file.txt.

Читання з файла і запис у файл

12.101. Написати програму, яка зчитує ціле число з файла input.txt і запише його в файл output.txt.

12.102. Написати програму, яка зчитує рядок символів з файла input.txt і запише його в файл output.txt.

12.103. Написати програму, яка зчитує за файла input.txt два цілих числа, записаних у стовпчик, обчислює їх найбільший спільний дільник і запише результат у файл output.txt.

12.104. Земельна ділянка має вигляд прямокутника зі сторонами a і b . Ці величини записано у файл start.txt. Програма повинна обчислити довжину діагоналі і записати її у файл finish.txt.

12.105. Файл input.txt містить два цілих числа a і b – межі діапазону. Програма повинна згенерувати випадкове число у цьому діапазоні і записати його в файл output.txt.

12.106. Написати програму, яка зчитує з файла input.txt рядок символів, видаляє з нього перші 6 символів і останні 3 символи, після чого запише рядок у файл output.txt.

12.107. Файл field.txt містить два масиви цілих чисел, записаних відповідно у два стовпчики: розміри полів прямокутної форми. Програма повинна обчислити сумарну площу всіх полів і записати результат у файл sq.txt.

12.108. Повідомлення, записане символами англійського алфавіту, шифрується таким чином: кожен символ повідомлення зсувається вправо на два. Якщо отриманий символ виходить за межі алфавіту, то відлік повинається спочатку. Програма повинна читати повідомлення з файла input.txt і записати зашифроване повідомлення у файл output.txt.

12.109. Файл start.txt містить межу діапазону y . Програма повинна згенерувати випадкове число у діапазоні від y до y^2 і записати його у файл output.txt.

12.110. Файл input.txt містить дві пари чисел, записаних у стовпчик: a_1 і b_1 , a_2 і b_2 . Довжина прямокутника являє собою випадкове число у діапазоні $[a_1;b_1]$, ширина прямокутника –

випадкове число у діапазоні $[a_2;b_2]$. Програма повинна обчислити площу прямокутника і записати її у файл output.txt.

12.111. Файл input.txt містить ціле число x . Радіус кола являє собою випадкове число у діапазоні $[1;x]$. Написати програму, яка обчислить площу круга і запише результат у файл output.txt.

12.112. Файл ro.txt містить чотири дійсних числа, записані у стовпчик: ro (густина речовини), a , b , c (розміри тіла у формі прямокутного паралелепіпеда). Програма повинна обчислити масу тіла і записати її у файл masa.txt.

12.113. Файл input.txt містить 3 дійсних числа, записані у стовпчик – урожайність поля і розміри поля (яке має форму прямокутної ділянки). Програма повинна обчислити урожай, зібраний з поля, і записати його у файл output.txt.

12.114. У файл start.txt у рядок записано 10 цілих чисел. Програма повинна обчислити найбільше і найменше з них, після чого записати їх у файл finish.txt.

12.115. Файл exist.txt містить два цілих числа – x_1 і x_2 . Програма повинна записати у список 7 випадкових чисел у діапазоні $[x_1;x_2]$, після чого записати цей список в один рядок у файл solution.txt.

12.116. У файл start.txt записано ціле число x . Програма повинна згенерувати два випадкових числа y_1 і y_2 у діапазоні $[0;x]$. Далі програма повинна заповнити масив 10 випадковими числами у діапазоні $[y_1;y_2]$ і записати його у файл finish.txt – у стовпчик.

12.117. Файл input.txt містить два масиви дійсних чисел – опади за перші два тижня серпня. Написати програму, яка визначить дату, коли кількість опадів була найбільшою, і запише результат у файл output.txt.

12.118. Два масиви містять дані про витрату води споживачами в m^3 і записані у файл input.txt, кожен в окремому рядку. Програма повинна обчислити сумарну витрату води і записати результат у файл output.txt.

12.119. Файл start.txt містить 8 дійсних чисел – радіуси кіл. Написати програму, яка обчислить суму площ цих кіл і запише результат у файл solution.txt.

12.120. Файл input.txt містить два числа: кількість елементів масиву n і межу діапазону x . Написати програму, яка заповнює масив n випадковими числами у діапазоні від x до x^2 , а потім записує масив у файл output.txt.

12.121. Масив з 10 цілих чисел, записаний у вигляді рядка, міститься у файлі input.txt. Програма повинна визначити найбільший і найменший елементи масиву і записати їх у файл output.txt.

12.122. Масив з 16 цілих чисел, записаний у вигляді рядка, міститься у файлі input.txt. Програма повинна визначити суму елементів масиву і записати її у файл output.txt.

12.123. Файл input.txt містить два масиви цілих чисел, записані кожен у рядок. Перший масив містить довжину кожної з земельних ділянок прямокутної форми, другий – ширину кожної з цих ділянок. Програма повинна обчислити площі цих ділянок, записати їх у рядок у файл output.txt.

12.124. Файл input.txt містить два цілих числа a і b – межі діапазону. Програма повинна згенерувати масив з 12 випадкових чисел у цьому діапазоні і записати його в файл output.txt (в рядок).

12.125. Написати програму, яка скопіює вміст файла input.txt у файл output.txt.

12.126. Файл in.txt містить натуральні числа. Програма повинна переписати у файл out.txt всі парні числа.

12.127. Написати програму, яка зчитує з файла digit.txt два дійсних числа, обчислює їх суму їх квадратів і записує її у файл esc.txt.

12.128. Файл input.txt містить символи англійського алфавіту, записані у стовпчик. Програма повинна скопіювати всі букви, що позначають голосні звуки, у файл output.txt.

12.129. Файл input.txt містить символи англійського алфавіту, записані у стовпчик. Програма повинна скопіювати символи у зворотному порядку у файл output.txt.

12.130. Файл input.txt містить прізвища учнів, записані у стовпчик. Програма повинна переписати всі прізвища, які починаються на літеру A, у файл output.txt.

12.131. У файл input.txt записано у стовпчик прізвища учнів. Програма повинна зчитати їх, вілестортувати у порядку зростання і записати у файл output.txt. Використати бібліотеку numpy.

12.132. Файл digit.txt містить число x – кількість символів у рядку, який необхідно заповнити випадковими символами англійського алфавіту і записати цей рядок у файл strings.txt.

12.133. Файл start.txt містить рядок тексту, який складається з символів англійського алфавіту. Змінна x генерується випадково у межах від 0 до довжини рядка. Програма повинна зчитати цей рядок, вилучити з нього символ, що має індекс x , і записати результат у файл finish.txt.

12.134. Файл coder.txt містить рядок символів українського алфавіту, розташованих за алфавітом. Користувач вводить довжину повідомлення L . Програма повинна згенерувати рядок випадкових символів українського алфавіту довжиною L і записати його у файл fraza.txt.

12.135. У файл `input.txt` записано текст, що складається з одного рядка довжиною від 10 до 100 символів. Програма повинна розбити цей текст на частини по 4 символи у кожній і записати їх у стовпчик у файл `output.txt`. Для обробки даних використати бібліотеку `numpy`.

12.136. Файл `start.txt` містить рядок довжиною від 8 до 80 символів. Програма повинна перевірити довжину рядку на парність. Якщо довжина рядка непарна, то дописати в кінці 0, розбити рядок на буквосполучення по 2 символи, кожне буквосполучення розвернути у зворотньому порядку і записати їх у стовпчик у файл `coded.txt`. У програмі використати бібліотеку `numpy`.

12.137. У файл `input.txt` записано повідомлення на англійській мові, яке необхідно зашифрувати. Для цього після кожного символу вхідного повідомлення програма повинна дописати випадковий символ англійського алфавіту і записати утворений рядок у файл `output.txt`. Реалізувати програму: 1) через роботу з рядками; 2) за допомогою функцій і методів бібліотеки `numpy`.

12.138. Файл `input.txt` містить повідомлення на українській мові, яке необхідно зашифрувати. Програма повинна розбити повідомлення на групи по 4 символи, всередині кожної групи поміняти місцями перший і третій, другий і четвертий символи. Далі групи символів збираються в один зашифрований рядок, який слід записати у файл `output.txt`.

12.139. У файл `start.txt` записано двовимірний масив A розмірністю $b \times 2$, перший стовпчик якого містить дані про довжину прямокутників, а другий – про ширину прямокутників. Програма повинна обчислити їх загальну площу і результат записати у файл `output.txt`. У програмі використати бібліотеку `numpy`.

12.140. Шкільне подвір'я вимощене квадратною плиткою розмірами a х a см. Всього використано N плиток. Числа a , N записані у файл `plytka.txt`. Програма повинна обчислити площу шкільного подвір'я у m^2 і записати результат у файл `square.txt`.

12.141. Вхідний файл `input.txt` містить натуральне число N – кількість повідомлень. Програма повинна створити N повідомлень, кожне з яких складається з двох випадкових символів англійського алфавіту і випадкової цифри, і записує ці повідомлення у файл `output.txt`.

12.142. Вхідний файл `seria.txt` містить серію номера автомобіля, яка складається з двох або трьох символів. Програма повинна згенерувати N номерів, кожен з яких складається з випадкового чотиризначного числа і серії, і записати ці номери у файл `numbers.txt`, кожен номер в окремому стовпчику.

12.143. Написати програму для генерування серійних номерів для однієї з програм. Вхідний файл starts.txt містить шестизначне число x . Програма повинна згенерувати випадкове число у діапазоні від 100000 до x , а до нього дописати три випадкових символи англійського алфавіту. Результат слід записати у файл serial.txt.

12.144. Вхідний файл data.txt містить парне число N довжини повідомлення. Саме повідомлення утворюється за таким принципом: кожен символ з непарним номером являє собою випадкову літеру англійського алфавіту, а кожен символ з парним номером – випадкову десяткову цифру. Утворене повідомлення записується у файл output.txt.

12.145. Файл input.txt містить число N – довжину повідомлення. Програма повинна згенерувати повідомлення, довжиною N символів, причому повідомлення повинно починатися і закінчуватися випадковим двоцифровим числом, а простір між цими числами бути заповненим нулями. Утворене повідомлення повинно записуватися у файл text.txt.

12.146. Файл abc.txt містить алфавіт повідомлень, довжина якого наперед невідома. Програма повинна створити повідомлення з N випадкових символів цього алфавіту і записати його у файл output.txt. Число N задається користувачем.

12.147. Файл input.txt містить одноцифрове число N і двоцифрове число Z . Програма повинна зчитати ці числа, створити масив розмірністю $N \times N$ і заповнити його випадковими числами у діапазоні $[10; Z]$, а утворений масив записати у файл output.txt. Для створення і обробки двовимірного масиву підключити бібліотеку numpy.

12.148. Файл start.txt містить рядок повідомлення. Програма повинна зчитати цей рядок, подвоїти символи у ньому і записати утворений рядок у файл output.txt.

12.149. Файл input.txt містить натуральне число N . Програма повинна створити повідомлення, яке складається з N випадкових десяткових цифр, і записати повідомлення у файл output.txt.

12.150. У файл input.txt записано N рядків тексту. Програма повинна скоротити рядки вдвічі і записати утворені рядки в файл output.txt.

Список використаних і рекомендованих джерел

1. Босова Л.Л., Аквилянов Н.А., Кочергин И.О. Информатика, 8-9 классы. Начала программирования на языке Python. Дополнительные главы к учебникам. М.: БИНОМ. Лаборатория знаний, 2020. 96 с.
2. Васильев А.Н. Python на примерах. Практический курс по программированию. СПб.: Наука и Техника, 2016. 432 с.
3. Жуков Р.А. Язык программирования Python: практикум: учеб.пособие. М.: ИНФРА-М, 2019. 216 с.
4. Костюченко А.О. Основы програмування мовою Python: навчальний посібник. Ч.: ФОП Балихьна С.М., 2020. 180 с.
5. Матвійчук С.В., Жуковський С.С. Практикум програмування Python / C++ на e-olymp.com (збірник задач з рекомендаціями до їх розв'язання). Житомир: Вид-во ЖДУ, 2019. 235 с.
6. Мюллер Дж.П. Python для чайников. СПб.: ООО «Диалектика», 2019. 416 с.
7. Рабинович В. Python для детей. Анимация с черепашкой графикой. М.: Литрес, 2020. 62 с.
8. Федоров Д.Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. М.: Издательство Юрайт, 2017. 126 с.

Интернет-джерела

1. Практикум з програмування мовою Python. URL: <https://pythonexercises.rozh2sch.org.ua/>
2. Путівник мовою програмування Python. URL: <https://pythonguide.rozh2sch.org.ua/>
3. Online Python Code Editor. <https://pynative.com/online-python-code-editor-to-execute-python-code/>
4. Python Extension Packages for Windows. URL: <https://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>
5. Thonny: Python IDE for beginners. <https://thonny.org>

Додатки

Додаток А

Перелік вбудованих функцій мови Python

Функції перетворення типів

`bool(x)` – перетворення у логічний тип. Якщо `x` є хибним або пропущений, повертає `false`, у протилежному випадку – `true`.

`bytearray([джерело [,кодування [помилки]])` – перетворення у послідовність цілих чисел в межах `[0; 256]`. Якщо викликати цю функцію без аргументів вона повертає порожній масив байт.

`bytes([джерело [,кодування [помилки]])` – повертає об'єкт типу `byte`, який є незмінною послідовністю цілих чисел в межах `[0; 256]`. Аргументи такі самі, як і в `bytearray()`.

`complex(real[,imag])` – перетворення у комплексне число

`dict([object])` – перетворення у словник

`float(x)` – перетворення у дійсне число

`frozenset(<послідовність>)` – повертає незмінну множину

`int(<object>, <основа системи числення>)` – перетворення у ціле число

`list(<object>)` – створює список

`memoryview(<object>)` – створює об'єкт `memoryview`

`object()` – повертає об'єкт, який є базовим для всіх об'єктів

`range(<start=0>,stop,<step=1>)` – арифметична прогресія від `start` до `stop` з кроком `step`

`set(<object>)` – створює множину

`slice(<start=0>,stop,<step=1>)` – об'єкт зрізу від `start` до `stop` з кроком `step`

`str(<object>,<coding>,<mistakes>)` – рядкове представлення об'єкта. Використовує метод `__str__`

`tuple(object)` – перетворення у кортеж

Інші вбудовані функції

`abs(x)` – модуль числа `x`

`all(послідовність)` – повертає `true`, якщо всі елементи послідовності істинні (або якщо послідовність порожня)

`any(послідовність)` – повертає `true`, якщо хоча б один елемент послідовності істинний. Для порожньої послідовності повертає `false`

`ascii(object)` – повертає рядок, який містить представлення об'єкта, але заміняє не-ASCII-символи на екрановані послідовності

`bin(x)` – перетворює ціле число у двійковий рядок

`callable(x)` – повертає `true` для об'єкта чи функції, які підтримують виклик

`chr(x)` – повертає односимвольний рядок, код символу якого дорівнює `x`

`classmethod(x)` – представляє вказану функцію методом класу

`compile(source, filename, mode, flags=0, don't_inherit=False)` – компілює у програмний код, який потім може виконуватися функціями `eval` або `exec`. Рядок не повинен містити символів кінця рядка або нульових байтів

`delattr(object, name)` – видаляє атрибут з іменем `name`

`dir([object])` – список імен об'єкта (якщо об'єкт не задано, то список імен у поточній локальній області видимості)

`divmod(a, b)` – повертає частку і остачу від ділення `a` на `b`

`enumerate(iterable, start=0)` – повертає лічильник, який при кожному проході надає кортеж з номера і відповідного члена послідовності

`eval(expression, globals=None, locals=None)` – виконує рядок програмного коду

`exec(object[, globals[, locals]])` – виконує програмний код

`filter(function, iterable)` – повертає лічильник з тих елементів, для яких `function` повертає істину

`format(value[, format_spec])` – форматування (рядка)

`getattr(object, name, [default])` – вилучає атрибут об'єкта або `default`

`globals()` – словник глобальних імен

`hasattr(object, name)` – перевіряє наявність атрибута з іменем `name`

`hash(x)` – повертає `hash` вказаного об'єкта

`help([object])` – викликає вбудовану довідкову систему

`hex(x)` – перетворення цілого числа в шістнадцятковий рядок

`id(object)` – повертає адресу об'єкта (унікальний ідентифікатор)

`input([prompt])` – повертає рядок, введений користувачем.

`Prompt` – підказка користувачу

`isinstance(object, ClassInfo)` – перевірка того, чи є об'єкт екземпляром класу `ClassInfo`. Якщо це так, то повертає істину, якщо ні – хибність

`issubclass(class, ClassInfo)` – перевіряє, чи є `class` підкласом `ClassInfo`. Якщо це так, то повертає істину

`iter(x)` – повертає об'єкт лічильника

`len(x)` – повертає кількість елементів у вказаному об'єкті

`locals(x)` – словник локальних імен

`map(function, iterator)` – лічильник, який утворився після застосування до кожного елемента послідовності функції `function`

`max(iter[, args...][, key])` – максимальний елемент послідовності

`min(iter[, args...][, key])` – мінімальний елемент послідовності

`next(x)` – наступний елемент лічильника

oct(x) – перетворює ціле число в рядок у вісімковому форматі
open(file,mode="r",buffering=None, encoding=None, errors=None,newline=None, closed=True) – відкриває файл і повертає відповідний потік

ord(x) – код символу

pow(x,y[,r]) – дія, еквівалентна $(x**y)\%r$

reversed(object) – лічильник з розвернутого об'єкта

repr(object) – представлення об'єкта

print([object,...],*,sep=" ", end='\n', file=sys.stdout) – друк

property (fget=None, fset=None, fdel=None, doc=None) – властивість

round(x, [,N]) – округлення до N знаків після коми

setattr(object, name, value) – встановлює атрибут об'єкта

sorted(iterable[,key][,reverse]) – сортує список

staticmethod(function) – статичний метод для функції

sum(iter,start=0) – сума членів послідовності

super([type[, object or type]]) – доступ до батьківського класу

type(object) – повертає тип об'єкта

type(name,bases,dict) – повертає новий екземпляр класу name

vars([object]) – словник з атрибутів об'єкта (за замовчуванням – словник локальних імен)

zip(*iters) – лічильник, який повертає кортежі, що складаються з відповідних елементів аргументів-послідовностей