

Лабораторна робота №6

Програмування вкладених циклів. Побудова таблиці значень функції двох змінних та опрацювання двовимірних масивів.

Мета роботи: ознайомлення оволодіння практичними навичками програмування алгоритмів циклічної структури, зберігання та опрацювання даних у формі двовимірних масивів, а також використання функцій форматного вводу та виводу числових даних.

Короткі теоретичні відомості

Під двовимірним масивом розуміють сукупність елементів одного типу, яка індексується за двома напрямками двома цілими числами та має спільну назву. Така сукупність даних має зручне візуальне подання у вигляді прямокутної таблиці, що відобразилося у вживанні термінів «рядки» чи «стовпці» до елементів з однаковим значенням одного з індексів (див. малюнок).

Стовпці 1...k				
Рядки 1...n	елемент(1,1)	елемент(1,2)	...	елемент(1, k)
	елемент(2,1)	елемент(2,2)	...	елемент(2, k)
	...	...	...	...
	елемент(n, 1)	елемент(n, 2)	...	елемент(n, k)

Доступ до конкретного елемента двовимірного масиву здійснюється вказанням двох чисел, – номера «рядка», та номера «стовпця». За значенням лише одного з індексів тут можна встановити лише сукупність елементів, яка сама по собі також є масивом, але вже одновимірним.

З точки зору збереження даних пам'яті комп'ютера така структура є дещо «штучною», бо адресація комірок пам'яті послідовною. Тому з двовимірним масивом працюють послідовно як із сукупністю одновимірних масивів, а в літературі з програмування часто можна зустріти визначення двовимірного масиву як масиву масивів або списку масивів.

Необхідність зберігання та програмного опрацювання двовимірних масивів обумовлюється, насамперед, їх прикладним значенням. В різних задачах фізики та техніки, економіки, нейромережевих технологіях систем штучного інтелекту і т. п. використовується математичний апарат матриць, які за своєю конструкцією і є двовимірними числовими масивами. Терміни матриця та двовимірний масив часто ототожнюються, насправді ж, поняття масиву є ширшим, оскільки позначає сукупність елементів довільного типу, а матриця утворюється лише з чисел.

Використання в **Python** списків замість масивів в двовимірному випадку завдає багато клопоту. Якщо в **Java** чи **C++** «масив масивів» – лише концепція, за якою система працює з багатовимірними масивами, то **список списків** в **Python**, – це реальний об'єкт, який потрібно створити в програмі для того, щоб зберігати та опрацьовувати двовимірний масив.

Система оголошення змінних в **Python**, яка визначає тип змінної за присвоєним їй значенням, хоча і є динамічною, проте строго контролює тип об'єкта записаного в змінну та дозволяє виконувати зі значенням змінної лише дії, які передбачені її типом. Інколи програмісти намагаються оголосити двовимірний список відповідних розмірів інструкцією вигляду:

```
<variable> = [[<value>] * <colCount>] * <rowCount>
```

ініціалізуючи всі його елементи деяким значенням.

Слушність такого підходу частково оправдана тим, що значення **value** має певний тип, і цей тип буде задавати тип елементів списку. Хоча динамічна типізація не буде перешкоджати запису в елемент такого списку значень будь-якого іншого типу. Оскільки списки в **Python** можуть містити об'єкти будь-якого типу, то для двовимірного масиву можна оголосити звичайний (одновимірний) список, а потім, в циклі, для кожного рядка масиву створити окремий список, та заповнити його елементами. У підсумку звертання до елемента матриці за індексами **p[i][j]** насправді конвертуватиметься у вибір **j-го** елемента в **i-му** списку-рядку.

Завдання 1.

Скласти програму для розв'язання наступних задач:

- Побудувати таблицю значень функцій двох змінних у заданій прямокутній області із заданими кроками для змінних x та y (див. варіант завдання).
- Результати обчислень занести у наперед створений для цього двовимірний масив.
- Обчислити в утвореному масиві найбільше та найменше значення, а також середнє арифметичне усіх елементів.
- Знайдені найбільше, найменше та середнє значення вивести на консоль, після чого надрукувати усі елементи масиву у вигляді прямокутної таблиці.

Варіанти завдань:

№ Варіанта	Функція	Змінна x		Змінна y	
		Відрізок	Крок	відрізок	крок
1.	$u = 2x^2 + 3y^2$	[-1; 1]	0.4	[-1; 1]	0.1
2.	$u = \sin x + \cos y$	[-2; 2]	0.5	[-1; 1]	0.1
3.	$u = \sqrt{x^2 + 4y^2}$	[0; 3]	0.5	[-3; 1]	0.2
4.	$u = x^3 + y^2x$	[0; 1]	0.2	[0; 10]	0.5
5.	$u = 5\sin(x + y)$	[-3; 3]	1	[-2; 2]	0.2
6.	$u = 2x^3 + 3y^2$	[-2; 2]	0.4	[-1; 1]	0.1
7.	$u = \sin xy + \cos \frac{x}{y}$	[-2; 2]	0.4	[1; 3]	0.1
8.	$u = \sqrt{2xy + 4y^3}$	[0; 3]	0.5	[0; 2]	0.1
9.	$u = x^3y + 3y^3x$	[0; 1]	0.2	[0; 1]	0.05
10.	$u = 5\lg(x + y)$	[1; 3]	0.25	[0; 4]	0.2
11.	$u = x^3 + y^2x$	[0; 2]	0.4	[0; 5]	0.25
12.	$u = 10\sin(x - y)$	[-3; 3]	1	[-2; 2]	0.2
13.	$u = xy + 3y^3x$	[0; 2]	0.4	[-1; 1]	0.1
14.	$u = 5\lg(x + y)$	[0; 1]	0.2	[-1; 0]	0.05
15.	$u = 2x^3 - 3y^2$	[-2; 2]	0.5	[-1; 1]	0.2
16.	$u = \sqrt{(1 - x)^2 + (2 - y)^2}$	[0; 2]	0.4	[0; 3]	0.3

Приклад виконання завдання 1

Варіант № 16. Умова задачі: Побудувати таблицю значень функцій двох змінних $u = \sqrt{2x^2 + 4y^2}$ у прямокутній області $[0; 2] \times [0; 3]$ із кроком 0.4 для змінної x та 0.1 – для y . Результати обчислень занести у наперед створений для цього двовимірний масив. Обчислити в утвореному масиві найбільше та найменше значення, а також середнє арифметичне усіх елементів. Знайдені найбільше, найменше та середнє значення вивести на консоль, після чого надрукувати усі елементи масиву у вигляді прямокутної таблиці.

Для зберігання двовимірного масиву зі значеннями функції створимо звичайний (одновимірний) список, а в нього додаватимемо списки значень функції при окремому значенні змінної y та різних значеннях змінної x . Заповнення масиву поєднаємо з обчисленням та виводом таблиці значень функції. Далі перейдемо до пошуку мінімуму та максимуму в заповненому масиві за допомогою пари вкладених циклів. Код програми мовою Python

```
import math

# Створюємо порожній список, в який
# записуватимемо рядки матриці
M = []

y1 = 0; y2 = 3; hy = 0.3
x1 = 0; x2 = 2; hx = 0.4
ky = int((y2 - y1) / hy) + 1
kx = int((x2 - x1) / hx) + 1

# обчислюємо таблицю значень функції та виводимо
# її на екран
print('%10s' % ' ', end='')
x = x1

# виводимо рядок заголовків зі значеннями x
for i in range(0, kx):
    print('%10.2f' % x, end='')
    x += hx

print()

# зовнішній цикл по y
y = y1
for i in range(0, ky):
```

Лабораторний практикум з програмування на Python

```
# в першому стовпці таблиці виводимо значення y
print('%6.2f      %y, end='')

x = x1

tmp = [] # створюємо новий список для наступного рядка
for j in range(0, kx):
    u = math.sqrt((1-x)*(1-x) + (2-y)*(2-y))
    print('%10.3f'%u, end='')
    tmp.append(u)
    x+= hx

y += hy
print()
M.append(tmp)

# пошук максимуму та мінімуму матриці
s = 0; min = M[0][0]; max = M[0][0]
for i in range(0, ky):
    for j in range(0, kx):
        s += M[i][j]
        if M[i][j] < min : min = M[i][j]
        if M[i][j] > max : max = M[i][j]
print()
print('min = %8.3f'%min, end='')
print(',   min = %8.3f'%max, end='')
s /= (kx*ky)
print(',   avg = %8.3f'%s, end='')
```

Завдання. 2

Скласти програму розв'язання задачі відповідно до варіанта. У звіт подати:

- умову задачі варіанта;
- складений Python-код;
- результати обчислень для деякого набору вхідних даних.

Варіанти завдань:

1. Заповнити матрицю $A(5, 6)$ випадковими числами з відрізка $[-10; 10]$. В кожному рядку матриці поміняти місцями останній та мінімальний елементи відповідного рядка. Вивести на екран вихідну та перетворену матрицю.

2. Дано масив розмірності 5×5 , елементи якого натуральні числа. Вивести на дисплей вихідний масив. Дзеркально відобразити його елементи відносно бічної діагоналі. Вивести початковий масив та результат його перетворення на дисплей.

3. У заданій квадратній матриці $M(7, 7)$ замінити середній рядок масивом, утвореним з мінімальних елементів кожного стовпця матриці. Вивести задану та перетворену матрицю на екран.

4. Задано квадратну матрицю $Q(6, 6)$, заповнену випадковими числами з відрізка $[-5; 5]$. Замінити усі елементи матриці, розташовані під головною діагоналлю на максимальний елемент відповідного їм рядка. Вивести на екран вихідну та змінену матрицю.

5. Дано масив розмірності 7×6 , елементи якого натуральні числа. Вивести на дисплей вихідний масив. Переставити рядки так, щоб в першому стовпці елементи були впорядковані за спаданням. Результат вивести на дисплей.

6. Задано матрицю $P(6, 7)$, заповнену випадковими числами з відрізка $[-9; 18]$. Поміняти місцями перший стовпець матриці, зі стовпцем, що містить максимальний її елемент. Вивести на екран вихідну та змінену матрицю.

7. У заданій квадратній матриці $B(7, 7)$ замінити середній стовпець масивом, утвореним з максимальних елементів кожного рядка матриці. Вивести задану та перетворену матрицю на екран.

8. Задано матрицю $M(7, 5)$, заповнену випадковими числами з відрізка $[-7; 7]$. Поміняти місцями перший рядок матриці, з рядком, що містить максимальний її елемент. Вивести на екран вихідну та змінену матрицю.

9. Дано масив розмірності 6×6 , елементи якого натуральні числа. Вивести на дисплей вихідний масив. Повернути його на 90° за годинниковою стрілкою. Результат вивести на дисплей.

10. Заповнити матрицю $A(6, 6)$ випадковими числами з відрізка $[-9; 9]$. В кожному рядку матриці поміняти місцями перший та максимальний елементи відповідного рядка. Вивести на екран матрицю A та утворену з неї, після перестановки вказаних елементів, матрицю.

11. Задано квадратну матрицю $Q(7, 7)$, заповнену випадковими числами з відрізка $[-7; 7]$. Замінити усі елементи матриці, розташовані над головною діагоналлю на максимальний елемент відповідного їм стовпця. Вивести на екран вихідну та змінену матрицю.

12. Задану квадратну матрицю $P(5, 5)$, перетворити за правилом: рядок з номером n зробити стовпчиком з номером n і навпаки, де n – задане число, введене користувачем з клавіатури. Вивести задану та перетворену матрицю на екран.

13. Скласти програму для заповнення квадратної матриці розміру 6×6 цілими числами так, як показано на малюнку. Заповнену матрицю відобразити симетрично відносно головної діагоналі. Вивести на екран утворену матрицю, виконати її перетворення і відобразити на екрані його результат.

0	1	2	3	4	5
2	3	4	5	6	7
4	5	6	7	8	9
6	7	8	9	10	11
8	9	10	11	12	13
10	11	12	13	14	15

14. Розробити програму для утворення квадратної матриці $M(7, 7)$ та заповненням її цілими числами за правилом, що проілюстроване поданим тут малюнком. Заповнену матрицю повернути на 90° за годинниковою стрілкою. Вивести на екран утворену матрицю, виконати вказане перетворення і відобразити на екрані його результат.

1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1

15. Дано масив розмірності 6×7 , елементи якого натуральні числа. Вивести на дисплей вихідний масив. Переставити рядки так, щоб в останньому стовпці елементи були впорядковані за зростанням. Результат вивести на дисплей.

16. У заданій квадратній матриці $M(6, 6)$ замінити головну діагональ масивом, утвореним з максимальних елементів кожного рядка матриці. Вивести задану та перетворену матрицю на екран.

Приклад виконання завдання 2.

Варіант № 16. Умова задачі: У заданій квадратній матриці $M(6, 6)$ замінити головну діагональ масивом, утвореним з максимальних елементів кожного рядка матриці. Вивести задану та перетворену матрицю на екран.

Для демонстрації результату роботи програми слід відобразити на екрані матрицю в початковому стані, та вигляд цієї матриці після виконаних перетворень. Деколи в таких випадках намагаються використовувати дві окремі матриці, – одну для вхідних даних, другу для формування результату. Умовою задачі таке не передбачається, тому користуватимемося двома масивами.

Алгоритм розв’язування задачі (див. блок-схему) буде таким:

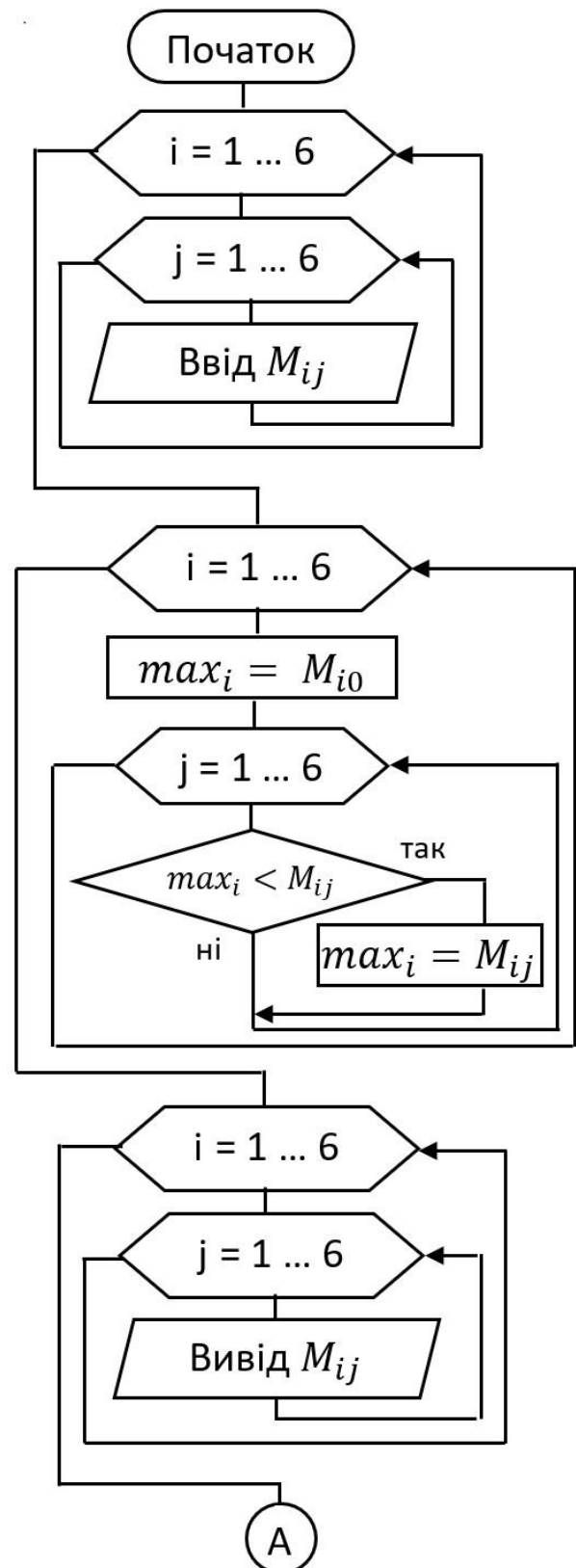
- спочатку програма отримує від користувача 36 значень, по одному для кожного елемента кожного рядка,;
- організовується цикл, що перераховує номери рядків матриці від 1 до 6;

- для i -го рядка внутрішній цикл шукає максимальний елемент, починаючи з першого та поступово порівнюючи з усіма наступними елементами цього рядка, кожного разу обираючи більший з двох;
- знайдений максимум i -го рядка записується в i -й елемент допоміжного одновимірного масиву максимумів;
- парою вкладених циклів виводимо елементи початкової матриці P по рядках;
- за допомогою одного циклу заміняємо усі елементи матриці з однаковими індексами на відповідний елемент з масиву максимумів.
- парою вкладених циклів виводимо по рядках елементи матриці після виконаної заміни.

Ввід усіх елементів двовимного масиву для опрацювання програмою, очевидно, займатиме в користувача багато часу. Якщо це відбувається в процесі експлуатації програми, то це цілком природньо, бо тут користувач вирішує поставлену задачу з допомогою створеної для цього програми.

Але цей процес неодмінно супроводжуватиме розробника програми у ході її налагодження, а це вже додаткові витрати часу. Тому в реалізації цього алгоритму ми заповнили (ініціалізувати) матрицю деякими вхідними даними, а під час виконання програми пропонуватимемо користувачеві або ввести елементи матриці, або виконати програму для заданих в програмі даних.

В кодї на **Python** ініціалізація змінних відіграє ключову роль, бо вона пов'язана з визначенням типу змінної. Масиви в **Python**, точніше списки, які ми використовуємо замість масивів, ініціалізують списком значень, розділеним комами, у квадратних дужках. Елементом списку може бути будь-який об'єкт, в



тому числі, і інший список. Ініціалізація двовимірного масиву в **Python** може виглядати так:

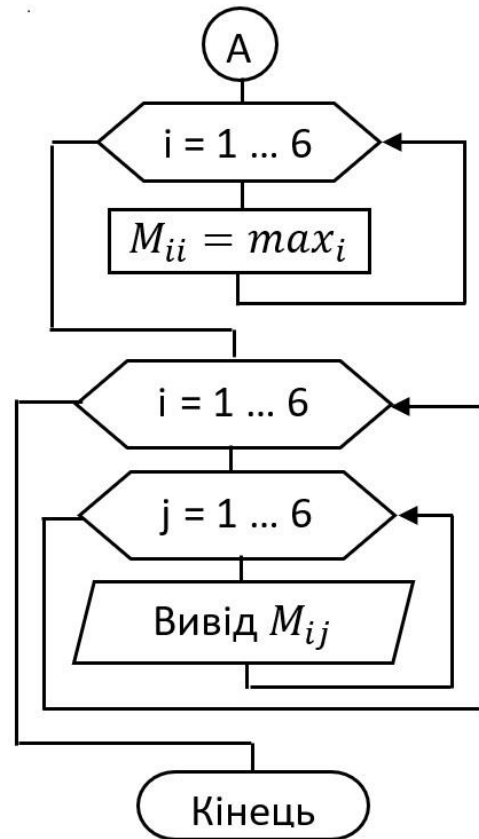
```
M = [[3, 4, 5], [3, 5, 7]]
```

В такий спосіб можна ініціалізувати як «прямокутні» так і «рвані масиви» – двовимірні масиви, в яких стрічки можуть містити різну кількість елементів.

Код програми для розв’язування задачі 2 на мові **Python**:

```
import array
size = 6
```

```
M = [[1,2,3,4,5,6], [1,3,5,7,9,1], [1,4,4,3,9,1], \
      [1,2,3,4,5,6], [2,2,2,3,3,5], [9,8,7,6,4,2]]
print('Бажаєте ввести матрицю з клавіатури?(так\ні)')
choise = input()
if choise == 'так':
    for k in range(size):
        tmp = array.array('i')
        for i in range(size):
            tmp.append(int(input()))
        M[k] = tmp
# Шукаємо максимальні елементи рядків
# і зберігаємо їх в окремому масиві
max = array.array('i')
for line in M:
    m = line[0]
    for x in line:
```



```

        if x > m: m = x
    max.append(m)
print('Матриця M:')
for line in M:
    for x in line:
        print('%3i'% x, end='')
    print()
#Виконуємо перетворення
for i in range(size):
    M[i].pop(i) # вилучаємо i-ий елемент з рядка, на його
    M[i].insert(i, max[i]) # місце вставляємо максимум
print('Результат перетворення:')
for line in M:
    for x in line:
        print('%3i'% x, end='')
    print()

```

Запитання для самоконтролю

1. Для чого в програмах використовують масиви (одновимірні)?
2. Які структури даних **Python** дозволяють опрацьовувати масиви?
3. Як оголосити список та проініціалізувати його елементами?
4. В чому полягає відмінність між списками та масивами?
5. Які синтаксичні конструкції **Python** дозволяють послідовно переглядати елементи списку?
6. Опишіть правила індексування елементів масиву.