

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ »

Варіант 14

Виконав студент: ІП-15 Кондрацька Соня Леонідівна

Перевірів: Вечерковська Анастасія Сергіївна

Лабораторна робота № 9 ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ

Мета – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Варіант 14

Задача

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Обчислення змінної, що описана в п.1, згідно з варіантом (табл. 1).

14	Задано матрицю дійсних чисел $A[n,n]$, ініціалізувати матрицю обходом по стовбцях. На головній діагоналі матриці знайти перший мінімальний і останній максимальний елементи, та поміняти їх з елементами головної діагоналі.
-----------	---

1) Постановка задачі

Ініціюємо двовимірний масив дійсного типу розміром $n \times n$. Заносимо в масив змінні обходом по стовбцям. За допомогою умов та арифметичного циклу знаходимо перший мінімальний та останній максимальний елементи головної діагоналі матриці та міняємо їх місцями. Результатом програми буде змінена початкова матриця.

2) Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Матриця	Дійсний	mas	Вхідні дані
Розмір матриці	Цілий	size	Вхідні дані
Матриця у підпрограмах	Дійсний	arr	Вихідні дані
Розмір матриці у підпрограмах	Цілий	n	Вхідні дані
Мінімальний та максимальний елементи головної діагоналі	Дійсний	min, max	Проміжні дані
Індекси мінімального та максимального елемента головної діагоналі	Цілий	imin, imax	Проміжні дані
Змінна для задання значення елементам матриці в порядку від k_0 до k	Дійсний	k	Проміжні дані
Прапор(використовується для обходу по стовбцям)	Цілий	xz	Проміжні дані

Лічильники арифметичних циклів	Цілий	i , j	Проміжні дані
Змінна для зміни елементів місцями	Дійсний	temp	Проміжні дані

В функції *input()* в масив *arr* заносимо елементи методом обходу по стовпцям(щоб бачити як саме заносяться данні в матрицю значення беруться від довільного дійсного числа *k* з кроком 1). За допомогою підпрограми *func()* знаходимо перший мінімальний та останній максимальний елементи головної діагоналі та їх індекси. Також одразу у цій функції міняємо їх місцями. В головній програмі ініціюємо матрицю та її розмір; викликаємо функції.

3) Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо роботу підпрограм.

Псевдокод

Основна програма:

Крок 1

Початок

Ініціювання масива

Виклик першої підпрограми

Виклик другої підпрограми

Кінець

Крок 2

Початок

double mas[size][size], size=6;

input(size, mas);

func(size, mas);

Кінець

Підпрограми:

input(int n, double arr[6][6])

double k=1.31, xz=-1;

для j від 0 до n повторити

якщо xz<0 то

для i від 0 до n повторити

a[i][j]=k;

k++;

все повторити

все якщо

інакше

для i від n-1 до 0 включно повторити

a[i][j]=k;

k++;

все повторити

xz=-xz;

все повторити

Кінець

func(int n, double arr[6][6])

double min=arr[0][0], max=arr[0][0];

int imin, imax;

для і від 0 до n повторити

якщо arr[i][i]<=min то

min=arr[i][i];

imin=i;

все якщо

якщо arr[i][i]>=max то

max=arr[i][i];

imax=i;

все якщо

все повторити

double temp;

temp=arr[imin][imin];

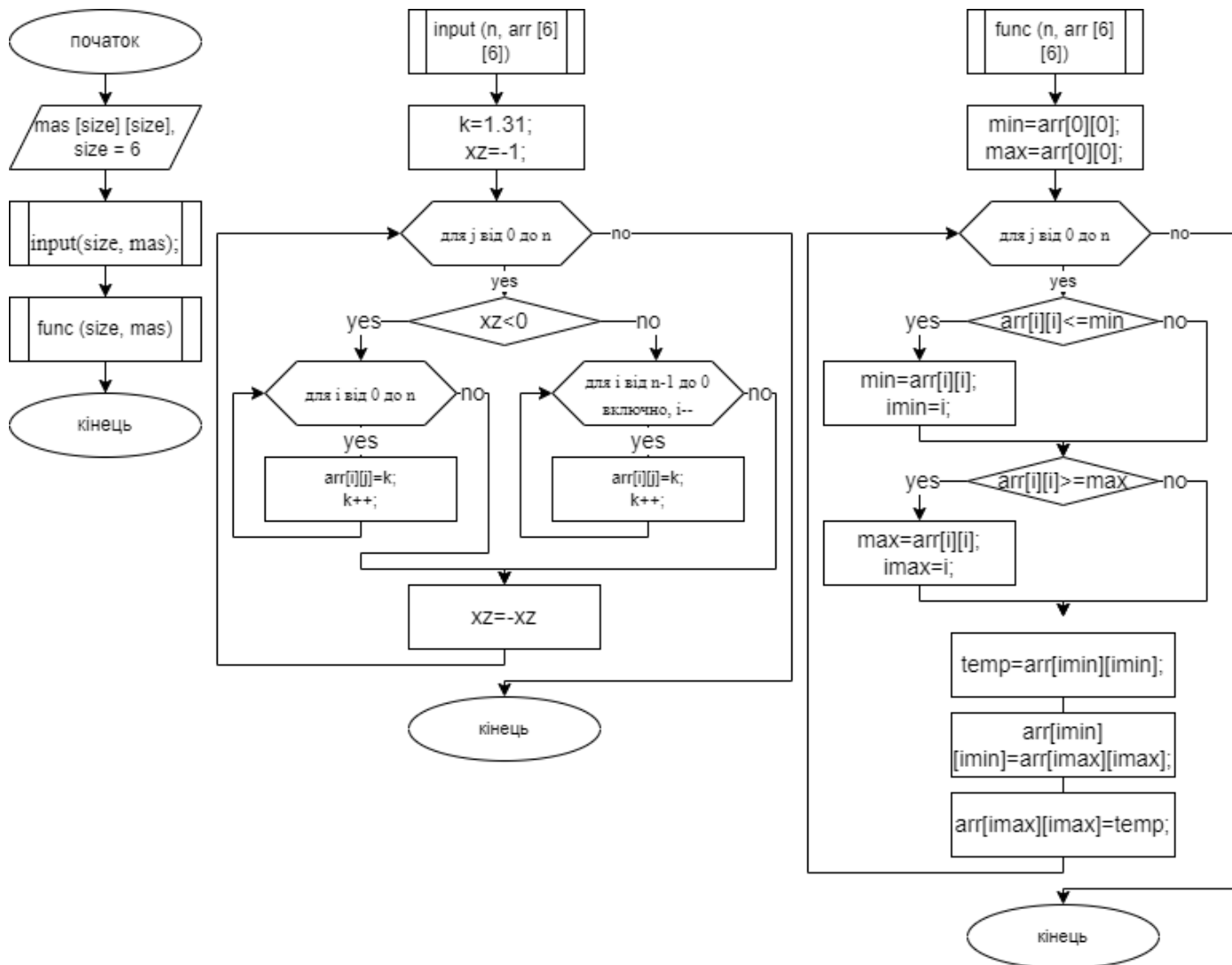
arr[imin][imin]=arr[imax][imax];

arr[imax][imax]=temp;

Кінець

4) Блок-схема

Основна програма: Підпрограми:



5) Код

```

1  #include <iostream>
2  using namespace std;
3  void input(int n, double arr[6][6])
4  {
5      double k = 1.31;
6      int xz = -1;
7      for (int j = 0; j < n; j++)
8      {
9          if (xz < 0)
10         {
11             for (int i = 0; i < n; i++)
12             {
13                 arr[i][j] = k;
14                 k++;
15             }
16         }
17         else
18         {
19             for (int i = n - 1; i >= 0; i--)
20             {
21                 arr[i][j] = k;
22                 k++;
23             }
24         }
25         xz = -xz;
26     }
27     cout << "Pochatkova matruzya" << endl;
28     for (int i = 0; i < n; i++)
29     {
30         for (int j = 0; j < n; j++)
31         {
32             cout << arr[i][j] << " ";
33         }
34         cout << endl;
35     }
36     cout << endl;
37 }
38 void func(int n, double arr[6][6])
39 {
40     double min = arr[0][0], max = arr[0][0];
41     int imin, imax;
42     for (int i = 0; i < n; i++)
43     {
44         if (arr[i][i] <= min)
45         {
46             min = arr[i][i];
47             imin = i;
48         }
49         if (arr[i][i] >= max)
50         {
51             max = arr[i][i];
52             imax = i;
53         }
54     }
55     double temp;
56     temp = arr[imin][imin];
57     arr[imin][imin] = arr[imax][imax];
58     arr[imax][imax] = temp;
59 }
60 void output(int n, double arr[6][6])
61 {
62     cout << "Zminena matruzya" << endl;
63     for (int i = 0; i < n; i++)
64     {
65         for (int j = 0; j < n; j++)
66         {
67             cout << arr[i][j] << " ";
68         }
69         cout << endl;
70     }
71 }
72 int main()
73 {

```

```

74 }
75 }
76 }
77 }
78 }
79 }
80 }

```

6) Випробування

Консоль отладки Microsoft Visual Studio

```

Pochatkova matruzya
1.31 12.31 13.31 24.31 25.31 36.31
2.31 11.31 14.31 23.31 26.31 35.31
3.31 10.31 15.31 22.31 27.31 34.31
4.31 9.31 16.31 21.31 28.31 33.31
5.31 8.31 17.31 20.31 29.31 32.31
6.31 7.31 18.31 19.31 30.31 31.31

Zminena matruzya
31.31 12.31 13.31 24.31 25.31 36.31
2.31 11.31 14.31 23.31 26.31 35.31
3.31 10.31 15.31 22.31 27.31 34.31
4.31 9.31 16.31 21.31 28.31 33.31
5.31 8.31 17.31 20.31 29.31 32.31
6.31 7.31 18.31 19.31 30.31 1.31

```

7) Висновки

Ми дослідили алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

В результаті виконання лабораторної роботи ми отримали програму для заповнення двовимірного масиву методом обходу по стовпцям, знаходження максимального та мінімального елементів головної діагоналі, а також зміна їх місцями, розділивши задачу на кроки: визначення основних дій, виклик першої підпрограми, виклик другої підпрограми.