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 20

Виконав студент ІП-15, Ликова Катерина Олександрівна
(шифр, прізвище, ім’я, по батькові)

Перевірив _____
(прізвище, ім’я, по батькові)

Київ 2021

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

Постановка задачі

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).

Таблиця 1

Задано матрицю дійсних чисел A[n,n], ініціалізувати матрицю обходом по рядках. На головній діагоналі матриці знайти перший максимальний і останній мінімальний елементи, а також поміняти їх місцями з елементами побічної діагоналі.

Математична модель

Змінна	**Тип**	**Ім'я**	**Призначення**
Кількість рядків і стовпців матриці	Цілий	n	Початкові дані
Матриця	Дійсний	A[][]	Результат
Масив, що містить середні значення елементів головної діагоналі матриці A	Дійсний	B[]	Проміжні дані
Функція, що надає матриці A значення елементів обходом по рядках	Порожній	input_matr	Проміжні дані
Функція, що виводить значення елементів матриці A	Порожній	output_matr	Проміжні дані
Функція, що створює масив B та знаходить його перший максимальний і останній	Порожній	task	Проміжні дані

мінімальний елементи, а також міняє їх місцями з елементами побічної діагоналі матриці A			
Змінна, що задає значення матриці A	Цілий	cnt	Проміжні дані
Змінна, що визначає парний рядок чи ні	Цілий	dir	Проміжні дані
Змінна, що слугує індексом масиву B від час його ініціалізації	Цілий	k	Проміжні дані
Максимальне значення елемента масиву B	Дійсний	max	Проміжні дані
Індекс першого максимального елемента масиву B	Цілий	l	Проміжні дані
Мінімальне значення елемента масиву B	Дійсний	min	Проміжні дані
Індекс останнього мінімального елемента масиву B	Цілий	m	Проміжні дані
Змінна для перестановки двох значень елементів матриці A	Дійсний	t	Проміжні дані
Лічильник циклів	Цілий	i	Проміжні дані
Лічильник циклів	Цілий	j	Проміжні дані
Параметр функцій input_matr, output_matr, task	Цілий	p1	Проміжні дані
Параметр функцій input_matr, output_matr, task	Цілий	p2	Проміжні дані
Параметр функції task	Дійсний	arr	Проміжні дані
Параметр функцій input_matr, output_matr, task	Дійсний	matr	Проміжні дані

Для вирішення даного завдання потрібно спочатку ввести значення змінної n. Далі використати функцію input_matr, де: присвоїти значення cnt, dir; виконати

цикл, де починаючи з одиниці якщо рядок непарний, заповнити його значеннями зліва направо збільшуючи кожен наступний елемент на 1; якщо рядок парний виконати аналогічні дії, але заповнюючи справа наліво. Якщо dir<0 рядок не парний, якщо dir<0 парний. Наступним кроком потрібно вивести матрицю A застосувавши функцію output_matr. Потім треба використати функцію task, де присвоїти значення k, за допомогою цикла присвоїти масиву B значення головної діагоналі матриці A, присвоїти значення max, знайти максимальне значення елемента B застосовуючи цикл, присвоїти значення l, знайти коефіцієнт першого максимального значення елемента B циклом, присвоїти значення min, знайти мінімальне значення елемента B застосовуючи цикл, присвоїти значення m, знайти коефіцієнт останнього мінімального значення елемента B циклом, замінюю отримані елементи відповідними того ж рядка побічної діагоналі матриці A. Виводимо отриману матрицю A функцією output_matr.

Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Вводимо значення змінної n.

Крок 3. Присвоюємо значення змінній A[][] та виводимо її.

Крок 4. Деталізуємо дії знаходження змінної A[][] та виводимо її.

Псевдокод

Основна програма

крок 1

початок

n

input_matr(A, n, n)

output_matr(A, n, n)

task(A, B, n, n)

output_matr(A, n, n)

кінець

Підпрограми

input_matr(matr, p1, p2)

```
cnt = 1
dir = -1
повторити для і від 0 до p1
        якщо dir < 0
                то
                        повторити для j від 0 до p2
                                A[i][j] = cnt+1
                        все повторити
        все якщо
        якщо dir > 0
                то
                        повторити для j від 1 до p2 - 1, j=j-1
                                A[i][j] = cnt+1
                        все повторити
        все якщо
dir = -dir
все повторити
кінець input_matr(matr, p1, p2)

output_matr(matr, p1, p2)
повторити для і від 0 до p1
        повторити для j від 0 до p2
                виведення matr[i][j]
        все повторити
все повторити
кінець output_matr(matr, p1, p2)

task (matr, arr, p1, p2)
k = -1
повторити для і від 0 до p1
```

```
        повторити для j від 0 до p2
            якщо i==j
                то
                    k = k+1
                    arr[k] = matr[i][j]
            все якщо
        все повторити
    все повторити
    max = 0
    повторити для i від 0 до p1
        якщо arr[i] > max
            то
                max = arr[i]
        все якщо
    все повторити
    l = -1
    повторити для i від 0 до p1
        якщо l < i та max == arr[i]
            то
                l = i
        все якщо
    все повторити
    min = max
    повторити для i від 0 до p1
        якщо arr[i] < min
            то
                min = arr[i]
        все якщо
    все повторити
    m = p1 + 1
```

повторити для i від 0 до p1

якщо m > i та max == arr[i]

то

m = i

все якщо

все повторити

повторити для i від 0 до p1

повторити для j від 0 до p2

якщо (i == l та j == l) або (i == m та j == m)

то

t = matr[i][j]

matr[i][j] = matr[i][p1 - 1 - j]

matr[i][p1 - 1 - j] = t

все якщо

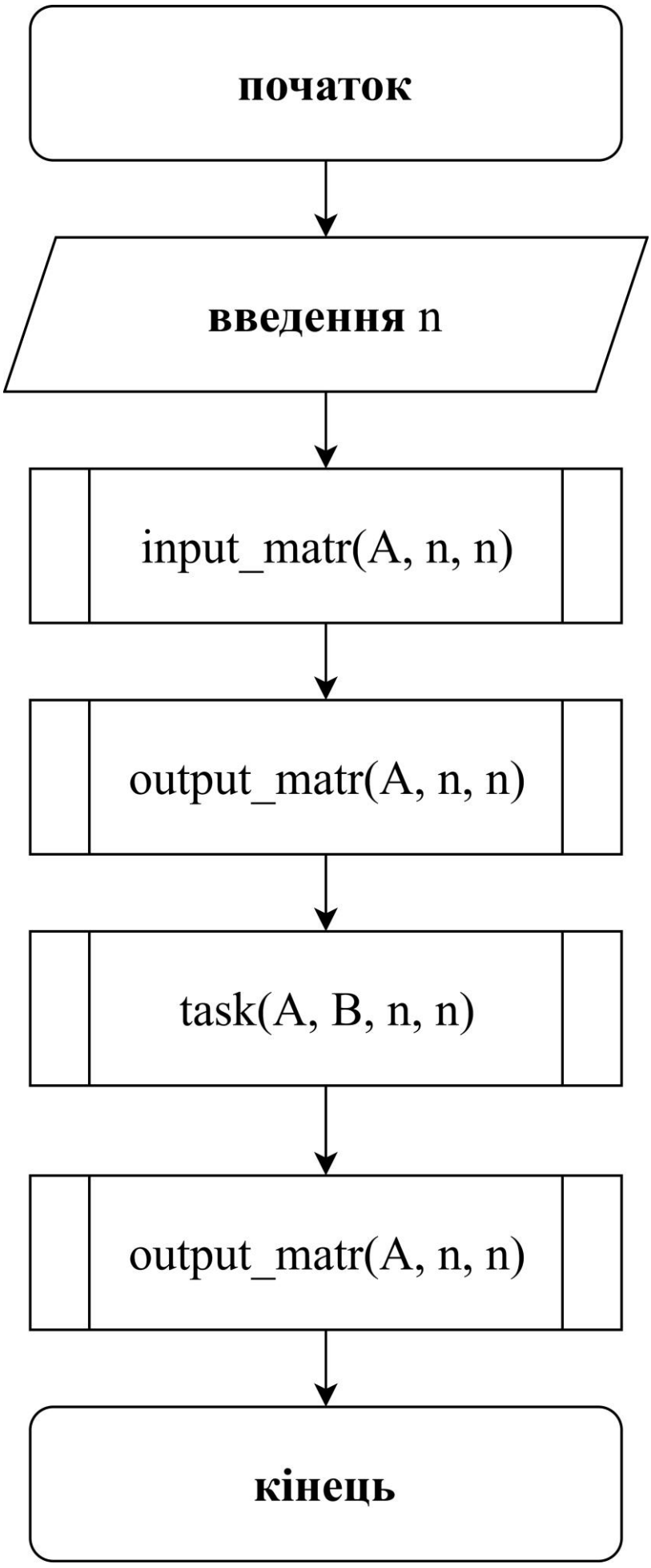
все повторити

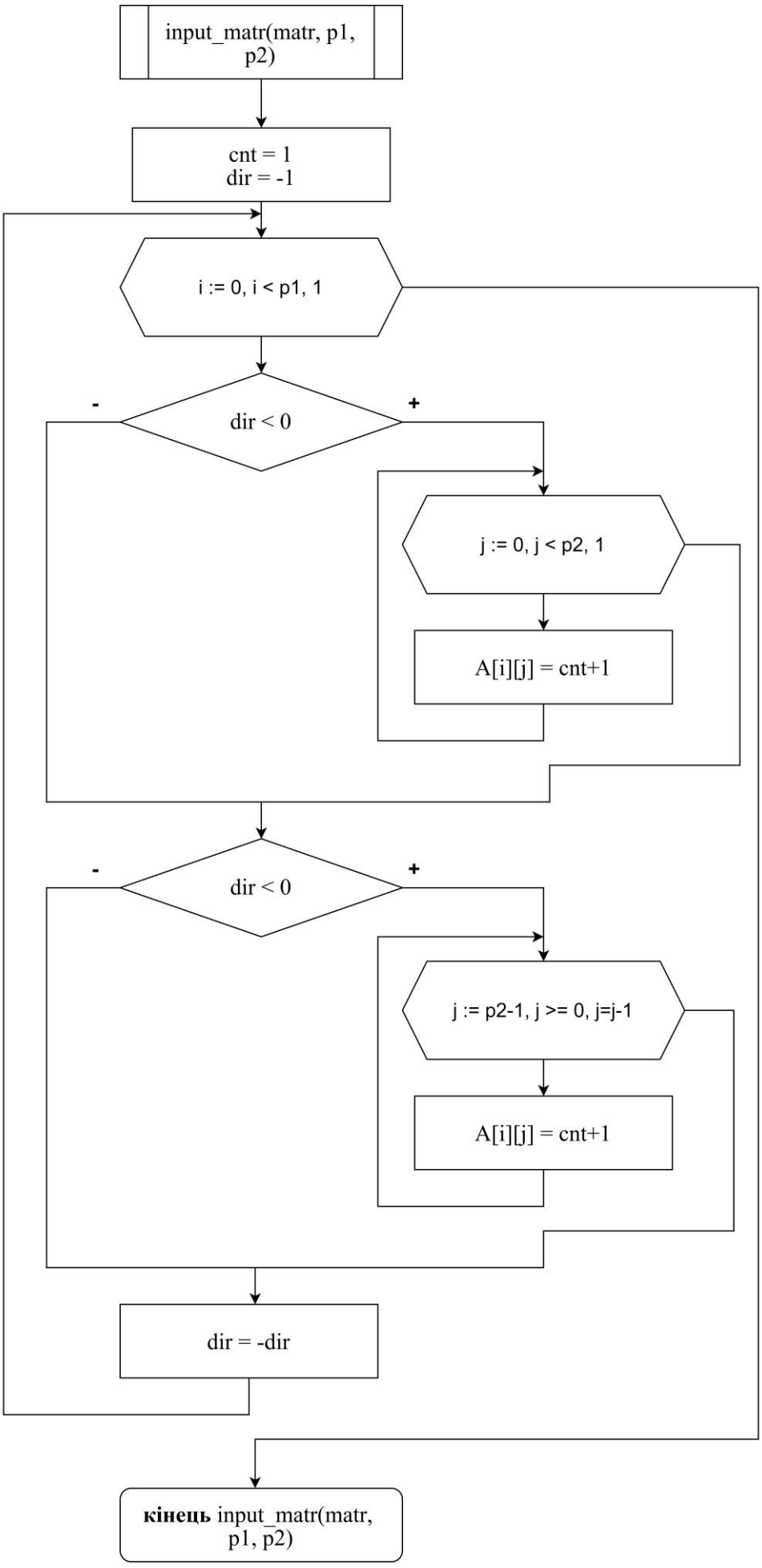
все повторити

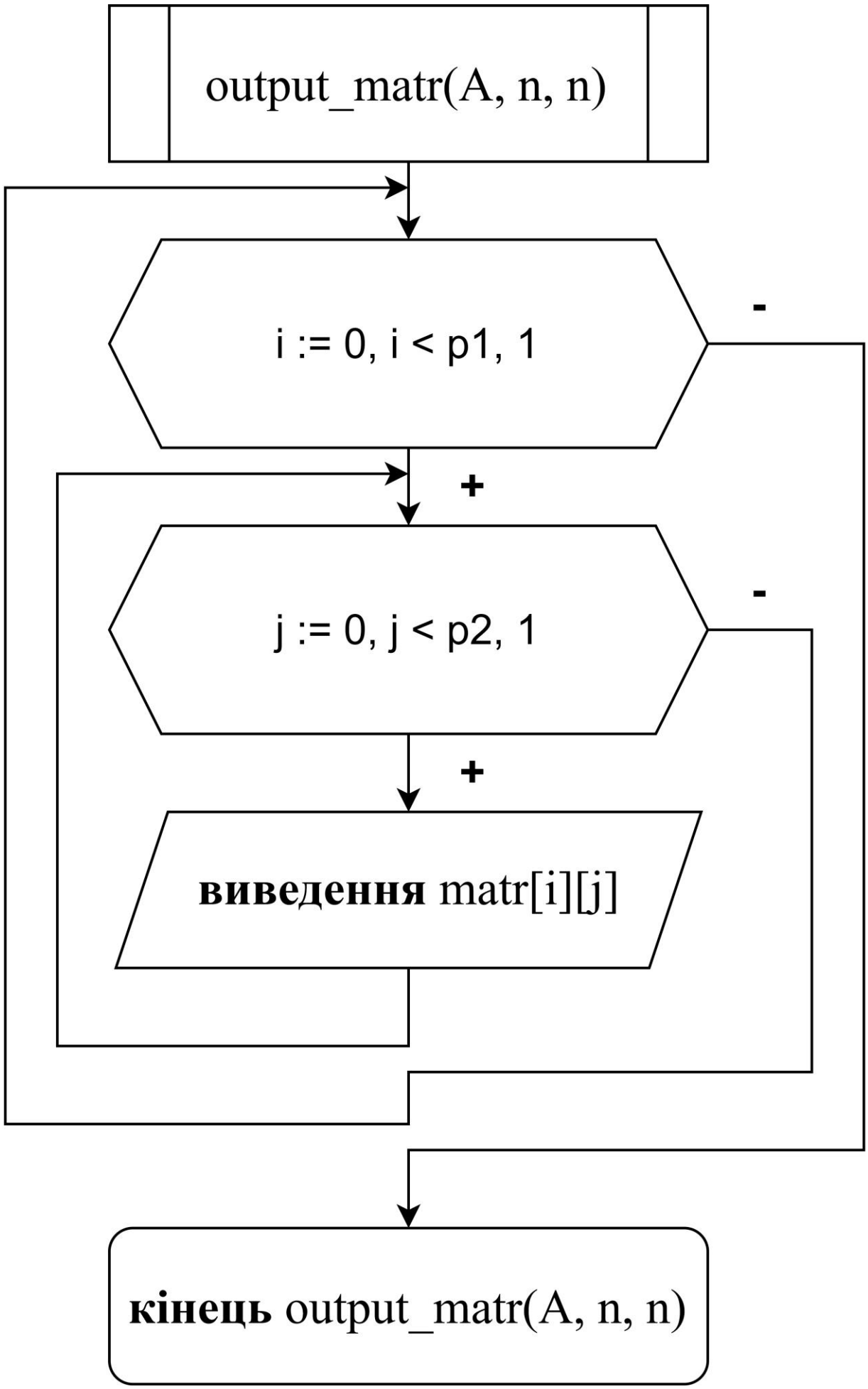
кінець task (matr, arr, p1, p2)

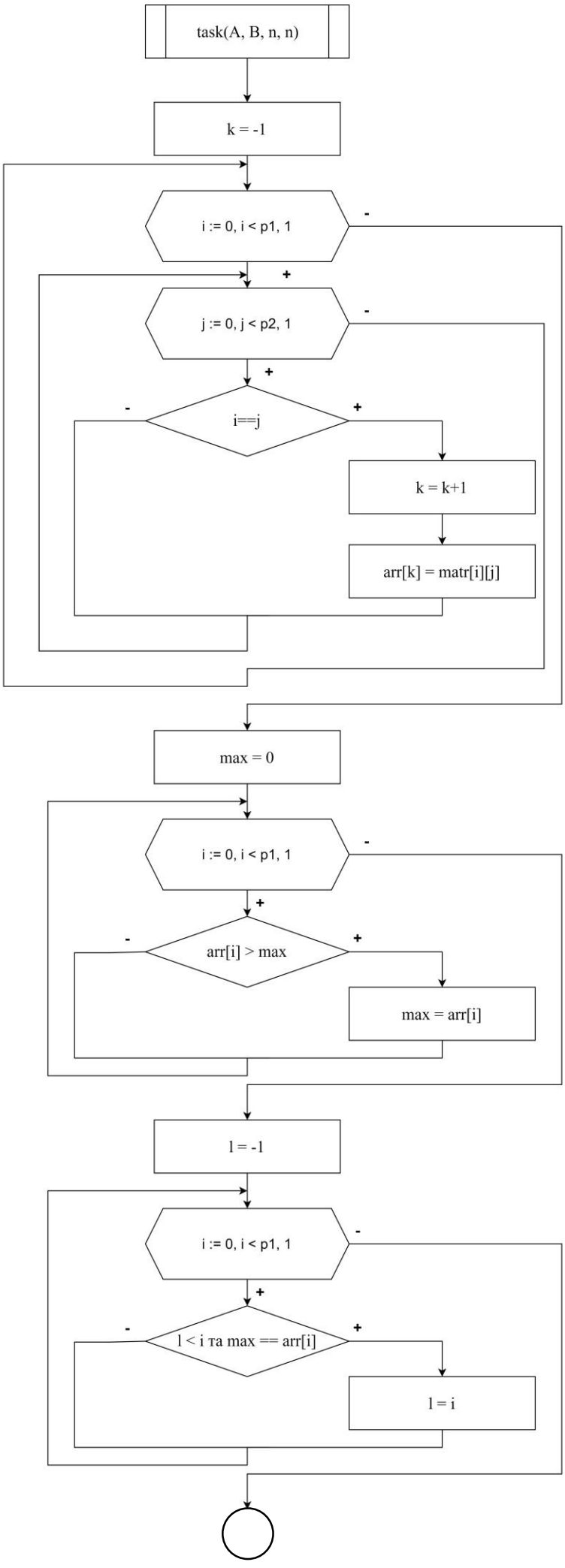
Блок-схема

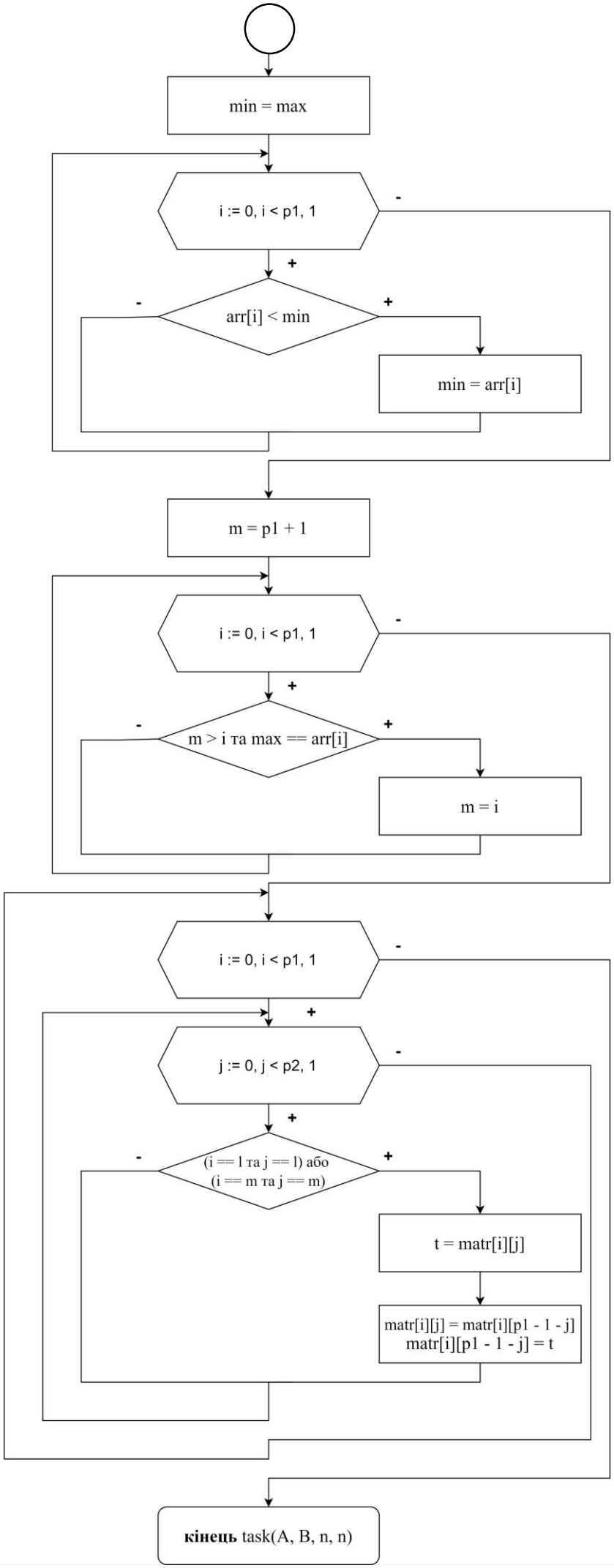
Основна програма


початок
введення n
input_matr(A, n, n)
output_matr(A, n, n)
task(A, B, n, n)
output_matr(A, n, n)
кінець


input_matr(matr, p1, p2)
cnt = 1
dir = -1
i := 0, i < p1, 1
-
dir < 0
+
j := 0, j < p2, 1
A[i][j] = cnt+1
-
dir < 0
+
j := p2-1, j >= 0, j=j-1
A[i][j] = cnt+1
dir = -dir
кінець input_matr(matr, p1, p2)


output_matr(A, n, n)
i := 0, i < p1, 1
-
+
j := 0, j < p2, 1
-
+
виведення matr[i][j]
кінець output_matr(A, n, n)


task(A, B, n, n)
k = -1
i := 0, i < p1, 1
-
+
j := 0, j < p2, 1
-
+
i==j
-
+
k = k+1
arr[k] = matr[i][j]
max = 0
i := 0, i < p1, 1
-
+
arr[i] > max
-
+
max = arr[i]
l = -1
i := 0, i < p1, 1
-
+
l < i та max == arr[i]
-
+
l = i


min = max
i := 0, i < p1, 1
-
+
arr[i] < min
-
+
min = arr[i]
m = p1 + 1
i := 0, i < p1, 1
-
+
m > i та max == arr[i]
-
+
m = i
i := 0, i < p1, 1
-
+
j := 0, j < p2, 1
-
+
(i == l та j == l) або
(i == m та j == m)
-
+
t = matr[i][j]
matr[i][j] = matr[i][p1 - 1 - j]
matr[i][p1 - 1 - j] = t
кінець task(A, B, n, n)

Код

```
#include <iostream>
#include <time.h>
#include <iomanip>
using namespace std;
typedef int Matrix[10][10];
typedef int Array[10];
Matrix A;
Array B;
void input_matr(Matrix, int, int);
void output_matr(Matrix, int, int);
void task(Matrix, Array, int, int);

int main()
{
srand(time(NULL));
int n;
cout << "input n: ";
cin >> n;
cout << "matrix A: " << endl;
input_matr(A, n, n);
output_matr(A, n, n);
task(A, B, n, n);
cout << "resulting matrix A: " << endl;
output_matr(A, n, n);
system("pause");
}
void input_matr(Matrix matr, int p1, int p2)
{
int cnt = 1;
```

```
int dir = -1;
for (int i = 0; i < p1; i++)
{
	if (dir < 0)
	{
		for (int j = 0; j < p2; j++)
		{
			A[i][j] = cnt++;
		}
	}
	if (dir > 0)
	{
		for (int j = p2 - 1; j >= 0; j--)
		{
			A[i][j] = cnt++;
		}
	}
	dir = -dir;
}

}
void output_matr(Matrix matr, int p1, int p2)
{
for (int i = 0; i < p1; i++)
{
	for (int j = 0; j < p2; j++)
	{
		cout << setw(5) << matr[i][j];
	}
	cout << endl;
```

```
}
cout << endl;
}
void task(Matrix matr, Array arr, int p1, int p2)
{
int k = -1;
for (int i = 0; i < p1; i++)
{
	for (int j = 0; j < p2; j++)
	{
		if (i == j)
		{
			k++;
			arr[k] = matr[i][j];
		}
	}
}
int max = 0;
for (int i = 0; i < p1; i++)
{
	if (arr[i] > max)
	{
		max = arr[i];
	}
}
int l = -1;
for (int i = 0; i < p1; i++)
{
	if (l < i && max == arr[i])
	{
```

```
                l = i;
        }
}
int min = max;
for (int i = 0; i < p1; i++)
{
        if (arr[i] < min)
        {
                min = arr[i];
        }
}
int m = p1 + 1;
for (int i = 0; i < p1; i++)
{
        if (m > i && min == arr[i])
        {
                m = i;
        }
}
for (int i = 0; i < p1; i++)
{
        for (int j = 0; j < p2; j++)
        {
                if ((i == l && j == l)||(i == m && j == m))
                {
                        int t = matr[i][j];
                        matr[i][j] = matr[i][p1 - 1 - j];
                        matr[i][p1 - 1 - j] = t;
                }
        }
```

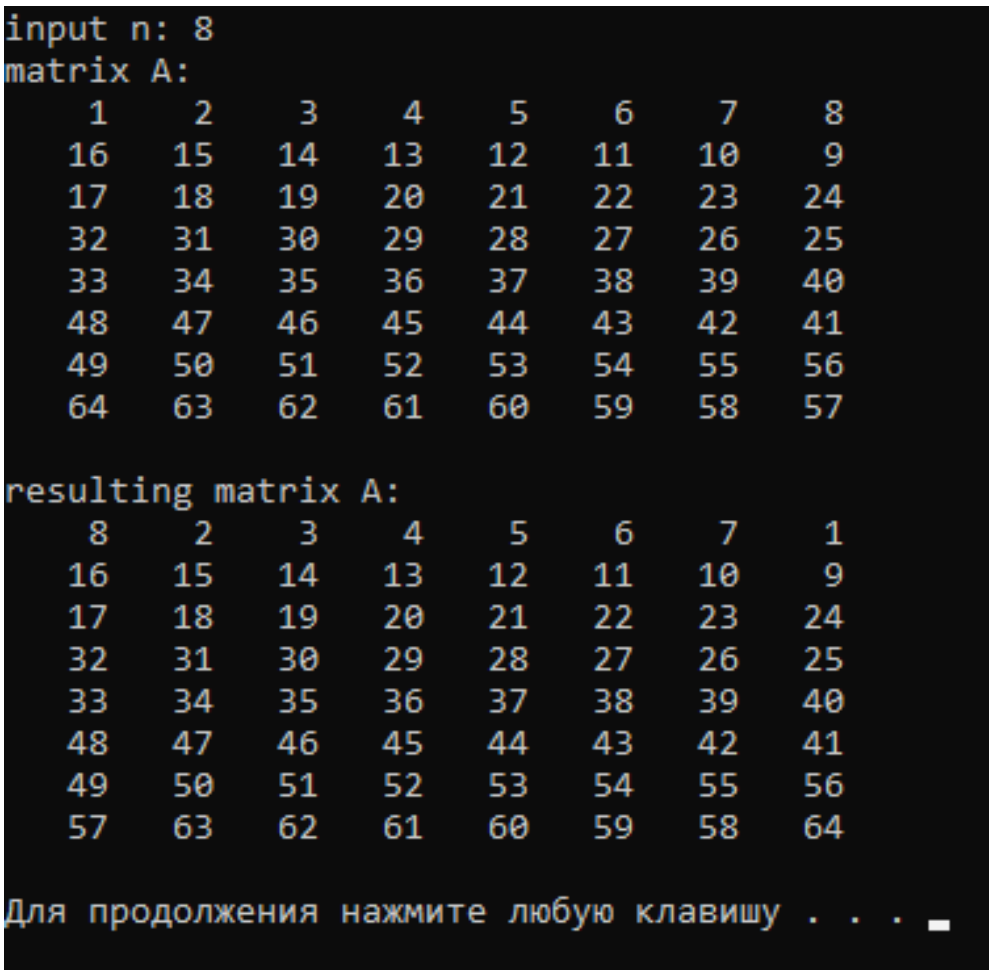
}

}

Тестування

```
input n: 10
matrix A:
    1    2    3    4    5    6    7    8    9   10
   20   19   18   17   16   15   14   13   12   11
   21   22   23   24   25   26   27   28   29   30
   40   39   38   37   36   35   34   33   32   31
   41   42   43   44   45   46   47   48   49   50
   60   59   58   57   56   55   54   53   52   51
   61   62   63   64   65   66   67   68   69   70
   80   79   78   77   76   75   74   73   72   71
   81   82   83   84   85   86   87   88   89   90
  100   99   98   97   96   95   94   93   92   91

resulting matrix A:
   10    2    3    4    5    6    7    8    9    1
   20   19   18   17   16   15   14   13   12   11
   21   22   23   24   25   26   27   28   29   30
   40   39   38   37   36   35   34   33   32   31
   41   42   43   44   45   46   47   48   49   50
   60   59   58   57   56   55   54   53   52   51
   61   62   63   64   65   66   67   68   69   70
   80   79   78   77   76   75   74   73   72   71
   81   82   83   84   85   86   87   88   89   90
   91   99   98   97   96   95   94   93   92  100

Для продолжения нажмите любую клавишу . . .
```

```
input n: 5
matrix A:
    1    2    3    4    5
   10    9    8    7    6
   11   12   13   14   15
   20   19   18   17   16
   21   22   23   24   25

resulting matrix A:
    5    2    3    4    1
   10    9    8    7    6
   11   12   13   14   15
   20   19   18   17   16
   25   22   23   24   21

Для продолжения нажмите любую клавишу . . .
```



Висновки: Для побудови алгоритму розв’язання заданої задачі я створила допоміжні функції з ітераційними алгоритмами обходу масивів по рядках для задання різних числових значень, знайшла потрібні значення на головній діагоналі та поміняла місцями з відповідними елементами побічної. Завдяки цьому я вивчила та дослідила властивості алгоритмів обходу масивів, отримала практичні навички використання цих алгоритмів під час складання програмних специфікацій.