

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 No 8 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 7

Виконав студент ІП-1407 Грицина Діана Русланівна (шифр, прізвище,  
ім'я, по батькові)

Перевірив Мартінова Оксана Петрівна ( прізвище, ім'я, по батькові)

Київ 2021

## Лабораторна робота 8

### Дослідження алгоритмів пошуку та сортування

**Мета** – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

#### Завдання

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. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом (табл. 1).

7	8 x 5	Дійсний	Із добутку значень елементів рядків двовимірного масиву. Відсортувати обміном за зростанням.
---	-------	---------	--

Постановка задачі: З використанням арифметичного циклу послідовно заповнюємо матрицю випадковими значеннями. Формуємо одновимірний масив та ініціюємо за умовою, поданою в завданні. Відсортуємо отриманий масив обміном за зростанням, послідовно порівнюючи кожний елемент з наступним.

Побудова математичної моделі

Змінна	Тип	Призначення
rows	int	Рядки у матриці
cols	int	Стовпці у матриці
matrix	double	Матриця
new_array	double	Новий одновимірний масив
Функція create_matrix	void	Ініціювання матриці
Функція create_array	void	Створення нового масиву на основі матриці
Функція sort_array	void	Сортування масиву обміном
Функція array_out	void	Виведення результату

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

- Крок 1. Визначимо основні дії
- Крок 2. Ініціювання матриці
- Крок 3. Ініціювання одновимірного масиву
- Крок 4. Сортування масиву
- Крок 5. Виведення елементів нового масиву

Псевдокод

Крок 1

**початок**

Ініціювання матриці

Ініціювання одновимірного масиву

Сортування масиву

Виведення елементів нового масиву

**кінець**

Крок 2

**початок**

rows = 8

cols = 5

matrix[rows][cols]

create\_matrix(matrix, rows, cols)

**функція create\_matrix(matrix, rows, cols)**

**повторити**

**для i від i1 до i2**

**повторити**

**для j від j1 до j2**

                matrix[i][j] = rand()

**все повторити**

**все повторити**

**кінець create\_matrix**

Ініціювання одновимірного масиву

Сортування масиву

Виведення елементів нового масиву

**кінець**

Крок 3

**початок**

rows = 8

cols = 5

```

matrix[rows][cols]
new_array[rows]
create_matrix(matrix, rows, cols)
create_array(matrix, rows, cols, new_array)
функція create_matrix(matrix, rows, cols)
повторити
    для i від i1 до i2
        повторити
            для j від j1 до j2
                matrix[i][j] = rand()
        все повторити
все повторити
кінець create_matrix
функція create_array(matrix, rows, cols, array)
повторити
    для i від i1 до i2
        mult = 1;
        повторити
            для j від j1 до j2
                mult *= matrix[i][j]
        все повторити
        arr[i] = mult
все повторити
кінець create_array
Сортування масиву
Виведення елементів нового масиву
кінець

```

Крок 4

```

початок
rows = 8
cols = 5
matrix[rows][cols]
new_array[rows]
create_matrix(matrix, rows, cols)
create_array(matrix, rows, cols, new_array)
sort_array(new_array, rows)
функція create_matrix(matrix, rows, cols)
повторити
    для i від i1 до i2
        повторити
            для j від j1 до j2
                matrix[i][j] = rand()

```

```

    все повторити
все повторити
кінець create_matrix
функція create_array(matrix, rows, cols, arr)
повторити
    для i від i1 до i2
        mult = 1;
        повторити
            для j від j1 до j2
                mult *= matrix[i][j]
        все повторити
        arr[i] = mult
все повторити
кінець create_array
функція sort_array(arr, n)
повторити
    для i від i1 до i2-1
        повторити
            для j від j1 до j2-1
                якщо arr[j]>arr[j+1]
                    то
                        tmp = arr[j]
                        arr[j] = arr[j+1]
                        arr[j+1] = tmp
                все якщо
        все повторити
все повторити
кінець sort_array
Виведення елементів нового масиву
кінець

```

Крок 5

**початок**

rows = 8

cols = 5

matrix[rows][cols]

new\_array[rows]

create\_matrix(matrix, rows, cols)

create\_array(matrix, rows, cols, new\_array)

sort\_array(new\_array, rows)

array\_out(new\_array, rows)

**функція create\_matrix(matrix, rows, cols)**

**повторити**

```

    для і від i1 до i2
        повторити
            для j від j1 до j2
                matrix[i][j] = rand()
            все повторити
все повторити
кінець create_matrix
функція create_array(matrix, rows, cols, arr)
повторити
    для і від i1 до i2
        mult = 1;
        повторити
            для j від j1 до j2
                mult *= matrix[i][j]
            все повторити
            arr[i] = mult
все повторити
кінець create_array
функція sort_array(arr, n)
повторити
    для і від i1 до i2-1
        повторити
            для j від j1 до j2-1
                якщо arr[j]>arr[j+1]
                    то
                        tmp = arr[j]
                        arr[j] = arr[j+1]
                        arr[j+1] = tmp
                все якщо
            все повторити
все повторити
кінець sort_array
функція array_out(arr, n)
повторити
    для і від i1 до i2
        виведення arr[i]
все повторити
кінець array_out
кінець

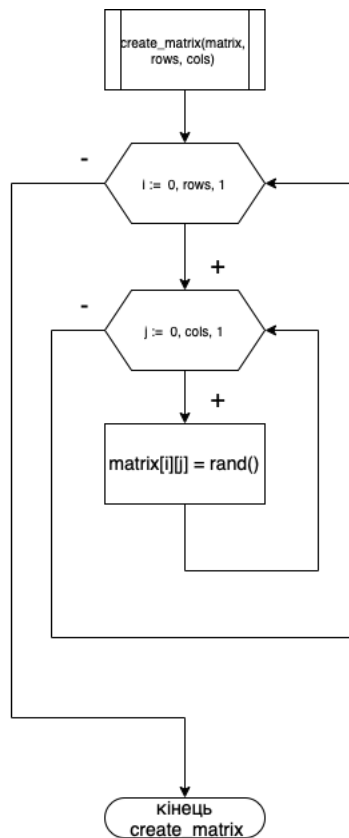
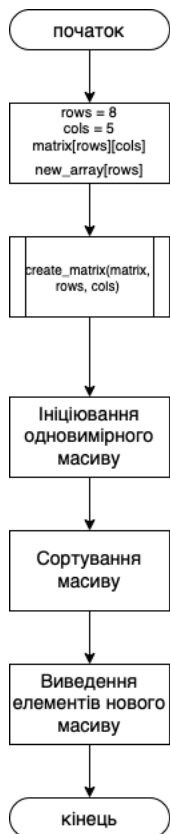
```

## Блок – схема

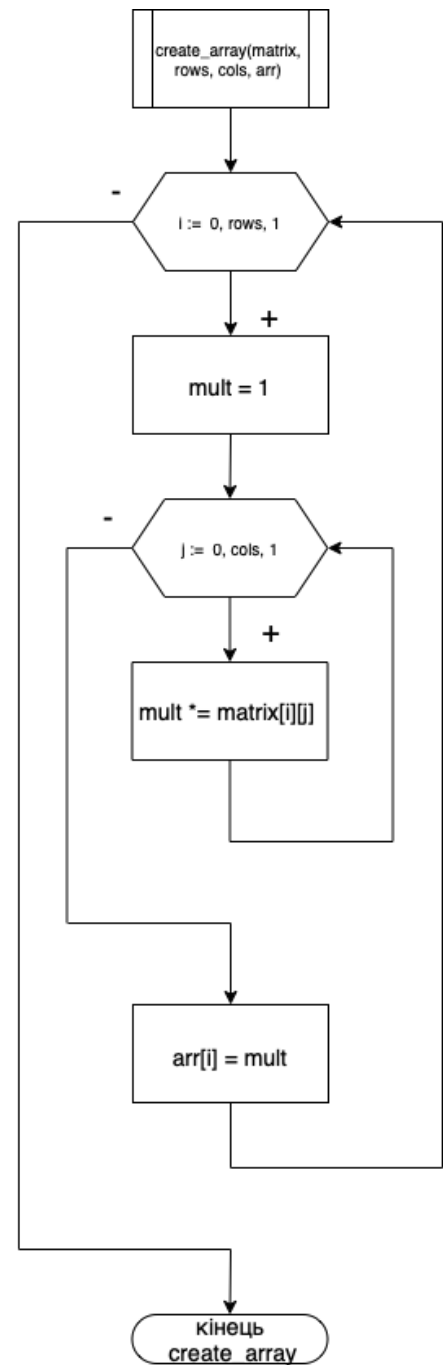
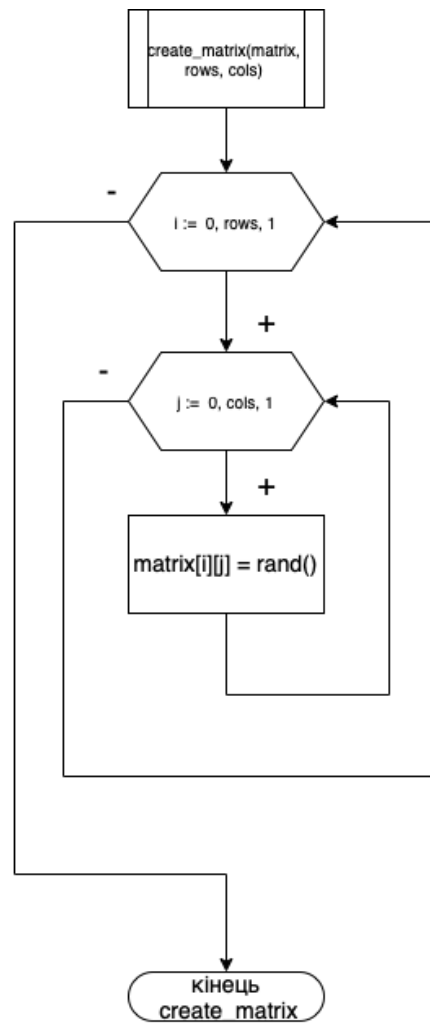
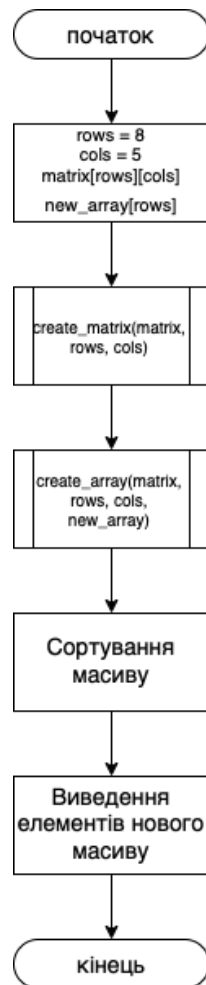
### Крок 1



### Крок 2

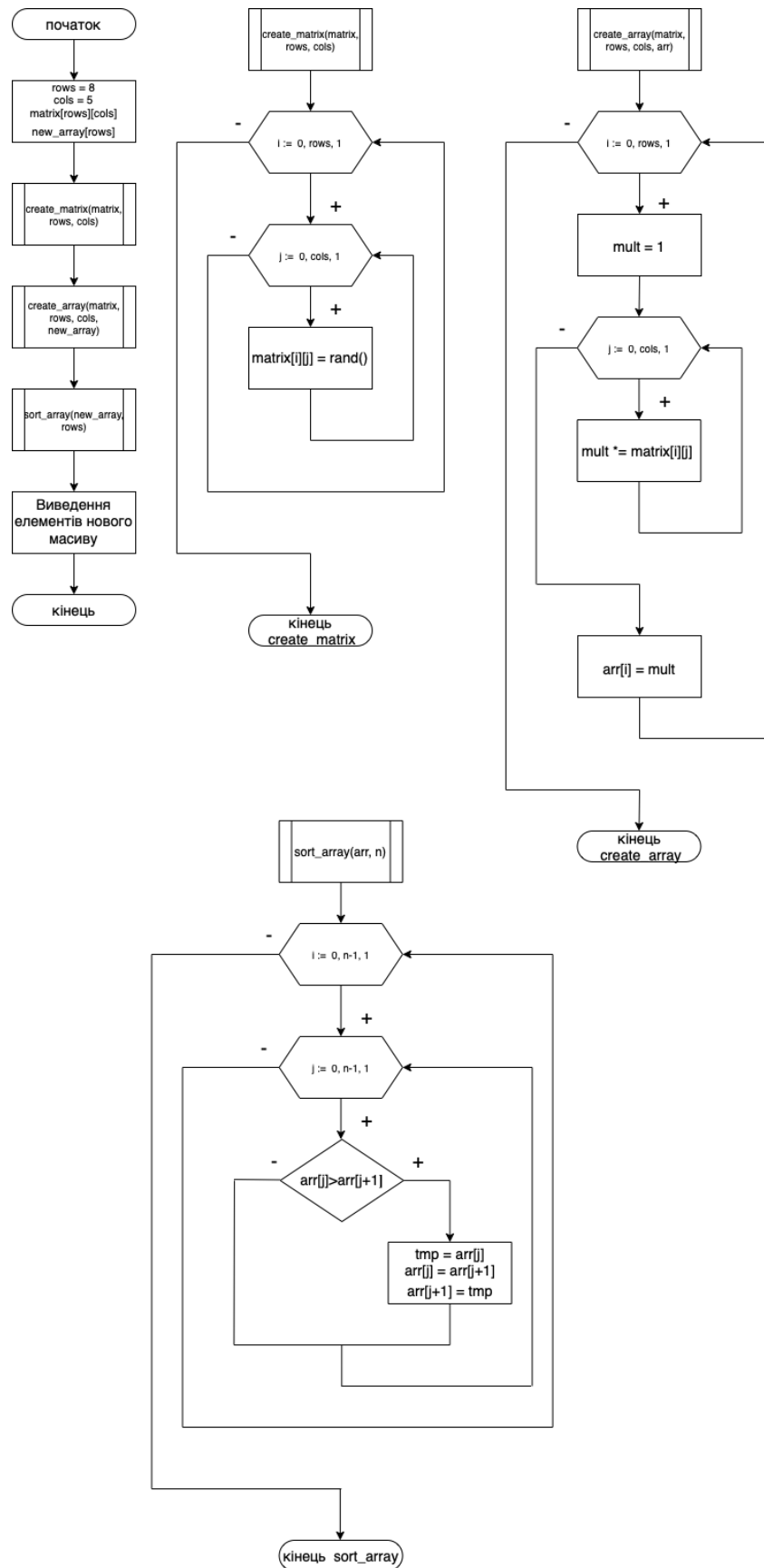


### Крок 3

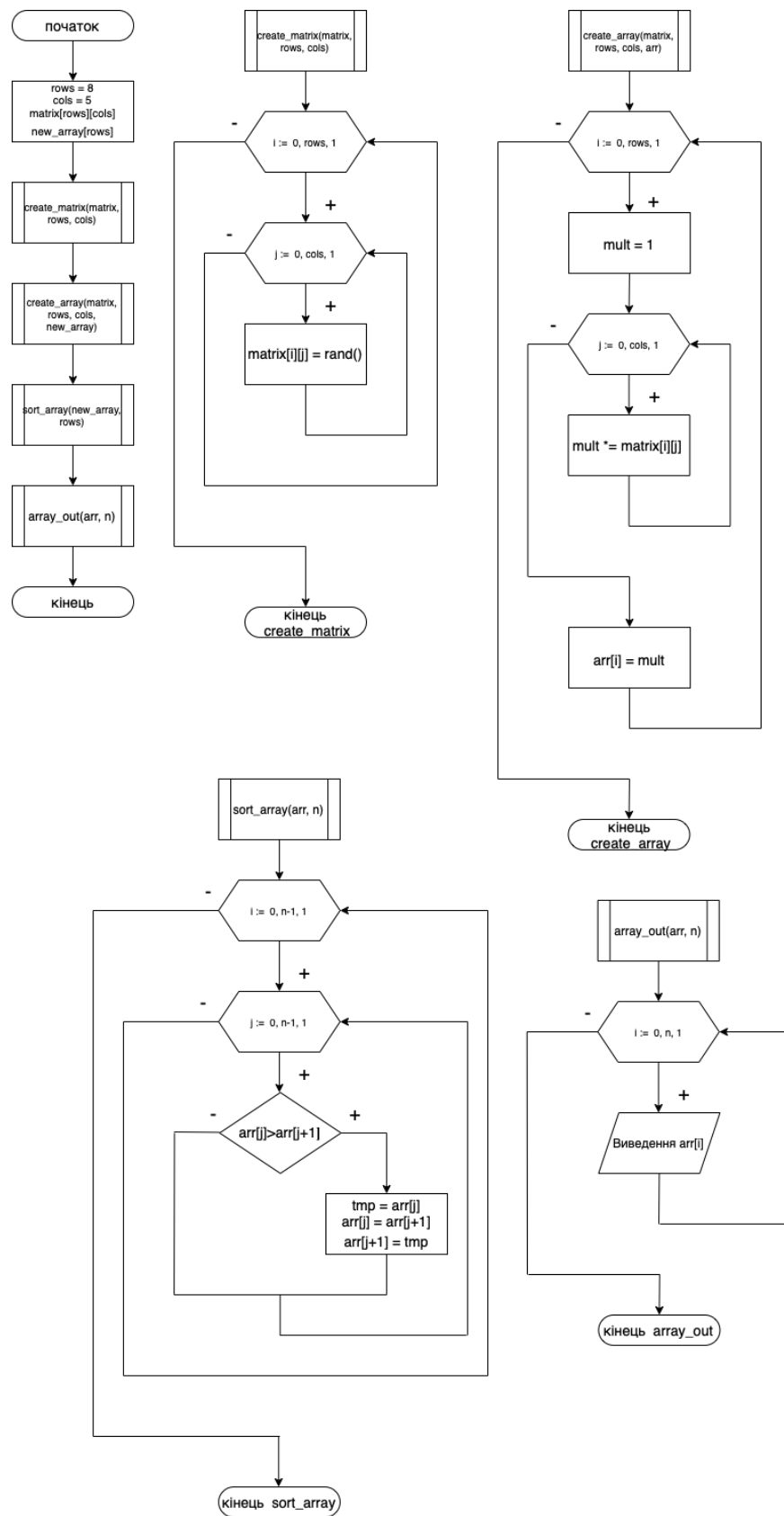




## Крок 4



## Крок 5



## Код програми на C++

```
#include <iostream>
using namespace std;
const int rows = 8;
const int cols = 5;
void create_matrix(double (&matrix)[rows][cols], int rows, int cols);
void create_array(double (&matrix)[rows][cols], int rows, int cols, double arr[]);
void sort_array(double arr[], int n);
void array_out(double arr[], int n);
int main(){
    double matrix[rows][cols];
    double new_array[rows];
    create_matrix(matrix, rows, cols);
    cout<<"\n"<<"New array"<<"\n";
    create_array(matrix, rows, cols, new_array);
    sort_array(new_array, rows);
    cout<<"\n"<<"Sorted array"<<"\n";
    array_out(new_array, rows);
}

void create_matrix(double (&matrix)[rows][cols],int rows, int cols){
    for (int i =0; i<rows; i++){
        for(int j = 0; j<cols; j++){
            matrix[i][j] = rand()%10 + 1;
            cout<<matrix[i][j]<<" ";
        }
        cout<<"\n";
    }
}

void create_array(double (&matrix)[rows][cols], int rows, int cols, double arr[]){

    for (int i =0; i<rows; i++){
        double multp = 1;
        for(int j = 0; j<cols; j++){
            multp *= matrix[i][j];
        }
        arr[i] = multp;
        cout<<arr[i]<<"\t";
    }
}
```

```
void sort_array(double arr[], int n){
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-1; j++){
            if(arr[j]>arr[j+1]){
                double tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
            }
        }
    }
}

void array_out(double arr[], int n){
    for(int i=0; i<n; i++){
        cout<<arr[i]<<"\t";
    }
    cout<<"\n";
}
```

## Результат

```
8 10 4 9 1
3 5 9 4 10
1 6 3 3 8
4 8 10 1 3
4 10 10 8 1
4 10 9 7 6
8 7 3 8 1
4 10 10 10 2
```

**New array**

**2880      5400      432 960 3200      15120      1344      8000**

**Sorted array**

**432 960 1344      2880      3200      5400      8000      15120**

## Перевірка

1 рядок =  $8 \cdot 10 \cdot 4 \cdot 9 \cdot 1 = 2880$

2 рядок =  $3 \cdot 5 \cdot 9 \cdot 4 \cdot 10 = 5400$

3 рядок =  $1 \cdot 6 \cdot 3 \cdot 3 \cdot 8 = 432$

4 рядок =  $4 \cdot 8 \cdot 10 \cdot 1 \cdot 3 = 960$

5 рядок =  $4 \cdot 10 \cdot 10 \cdot 8 \cdot 1 = 3200$

6 рядок =  $4 \cdot 10 \cdot 9 \cdot 7 \cdot 6 = 15120$

7 рядок =  $8 \cdot 7 \cdot 3 \cdot 8 \cdot 1 = 1344$

8 рядок =  $4 \cdot 10 \cdot 10 \cdot 10 \cdot 2 = 8000$

Отримуємо масив: 2880, 5400, 432, 960, 3200, 15120, 1344, 8000

В порядку зростання: 432, 960, 1344, 2880, 3200, 5400, 8000, 15120

**Висновок:** У роботі досліджено особливості алгоритму сортування обміном. Для обробки значень елементів матриці було використано два оператори повторення: один вкладений в другий. На основі елементів матриці було створено одновимірний масив, в якому всі сусідні елементи попарно порівнюються один з одним і міняються місцями, якщо попередній елемент більший від наступного. В результаті максимальний елемент поступово зміщується вправо і після першого такого перегляду займе крайнє праве положення. Потім процес перегляду повторюється і на останньому кроці отримаємо впорядковану послідовність.