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

«Дослідження алгоритмів розгалуження»

Варіант 17

Виконав студент ІП-12, Коновалюк Іванна Леонідівна

Перевірив _____
(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.

Варіант 17

	Розмірність	Тип даних	Обчислення значень одновимірного масиву
17	6 x 5	Цілий	Із суми додатних значень елементів рядків двовимірного масиву. Відсортувати методом вставки за зростанням.

Побудова математичної моделі.

Змінна	Тип	Ім’я	Призначення
Кількість рядків двовимірного масиву	Цілий	n	Початкові дані
Кількість стовпців двовимірного масиву	Цілий	m	Початкові дані
Двовимірний масив	Цілий	matrix	Проміжні дані
Тимчасова змінна для перестановки елементів	Цілий	t	Проміжні дані
Ітератор	Цілий	i	Проміжні дані
Ітератор	Цілий	j	Проміжні дані
Одновимірний масив	Цілий	array	Кінцеві дані

rand()%n – повертає, випатковим чином згенероване, ціле число з діапазону [0, n).

Власні функції

inputMatrix(двовимірний масив) – створює двовимірний масив
inputArray(двовимірний масив, одновимірний масив) – створює одновимірний масив
sortArray(одновимірний масив) – повертає одновимірний масив, елементи якого відсортовані за зростанням
outputMatrix(двовимірний масив) – виведить двовимірний масив
outputArray(одновимірний масив) – виводить одновимірний масив

Розв’язання

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію оголошення масивів.

Крок 3. Деталізуємо дію визначення двовимірного масиву за допомогою функції.

Крок 4. Деталізуємо дію визначення одновимірного масиву за допомогою функції.

Крок 5. Деталізуємо дію виведення двовимірного масиву.

Крок 6. Деталізуємо дію виведення одновимірного масиву.

Крок 7. Деталізуємо дію сортування одновимірного масиву.

Крок 8. Деталізуємо дію виведення відсортованого масиву.

Псевдокод алгоритму.

Крок 1.

Початок

Оголошення масивів

Визначення двовимірного масиву

Визначення одновимірного масиву

Виведення двовимірного масиву

Виведення одновимірного масиву

Сортування одновимірного масиву

Виведення відсортованого одновимірного масиву

Кінець

Крок 2.

Початок

n = 6

m = 5

matrix[n][m]

array[n]

Визначення двовимірного масиву

Визначення одновимірного масиву

Виведення двовимірного масиву

Виведення одновимірного масиву

Сортування одновимірного масиву

Виведення відсортованого одновимірного масиву

Кінець

Крок 3.

Початок

n = 6

m = 5

matrix[n][m]

array[n]

inputMatrix(matrix)

Визначення одновимірного масиву

Виведення двовимірного масиву

Виведення одновимірного масиву

Сортування одновимірного масиву

Виведення відсортованого одновимірного масиву

Кінець

Крок 4.

Початок

n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix(matrix)
inputArray(matrix, array)
Виведення двовимірного масиву
Виведення одновимірного масиву
Сортування одновимірного масиву
Виведення відсортованого одновимірного масиву
Кінець

Крок 5.

Початок
n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix(matrix)
inputArray(matrix, array)
outputMatrix(matrix)
Виведення одновимірного масиву
Сортування одновимірного масиву
Виведення відсортованого одновимірного масиву
Кінець

Крок 6.

Початок
n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix(matrix)
inputArray(matrix, array)
outputMatrix(matrix)
outputArray(array)
Сортування одновимірного масиву
Виведення відсортованого одновимірного масиву
Кінець

Крок 7.

Початок
n = 6

m = 5
matrix[n][m]
array[n]
inputMatrix(matrix)
inputArray(matrix, array)
outputMatrix(matrix)
outputArray(array)
sortArray(array)
Виведення відсортованого одновимірного масиву

Кінець

Крок 7.

Початок

n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix(matrix)
inputArray(matrix, array)
outputMatrix(matrix)
outputArray(array)
sortArray(array)
outputArray(array)

Кінець

Підпрограма inputMatrix(matrix)

```
повторити
для і від 1 до n
    повторити
    для j від 1 до m
        matrix[i][j] = rand()%100-0
    все повторити
все повторити
```

Все підпрограма

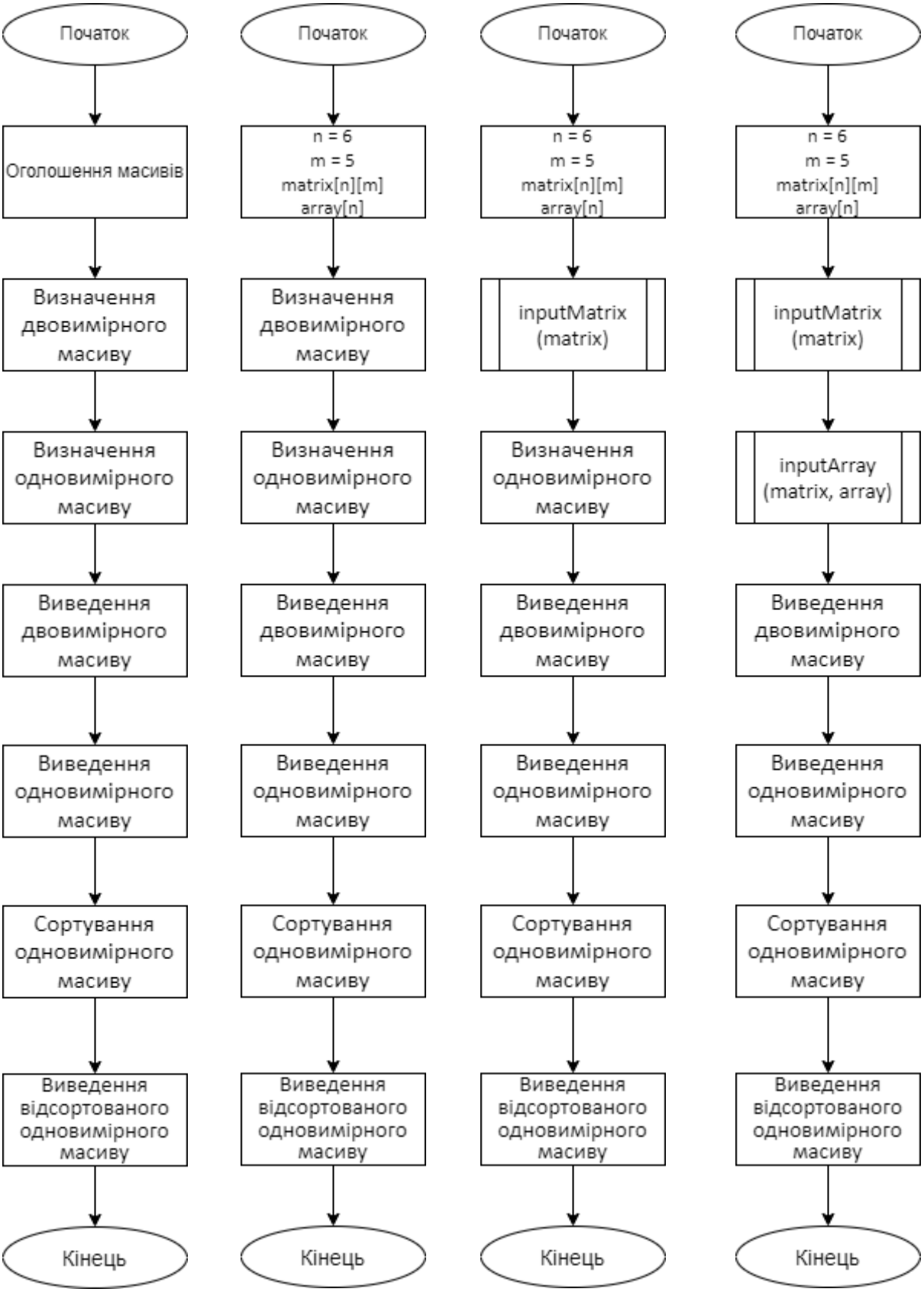
Підпрограма inputArray(matrix, array)

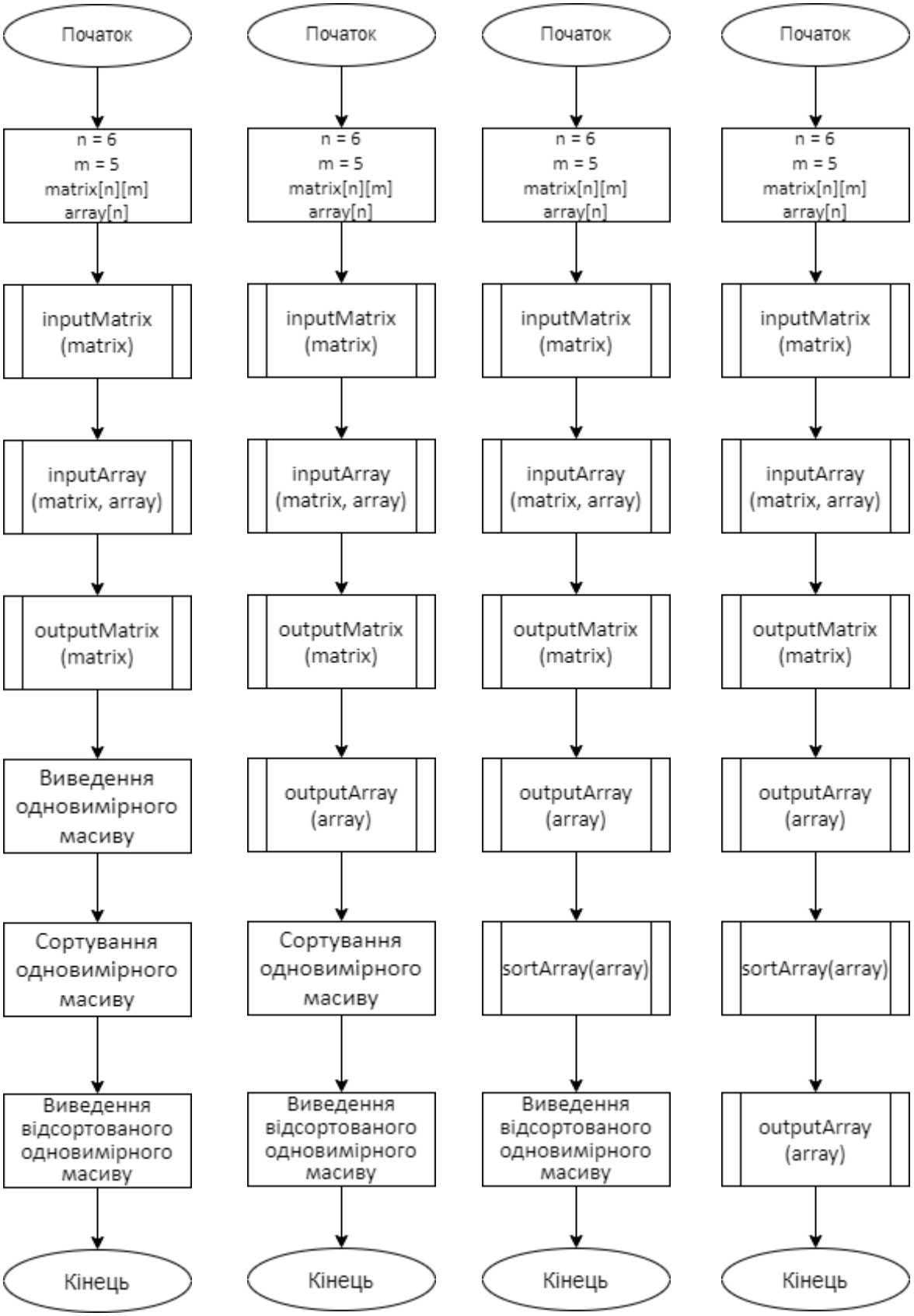
```
повторити
для і від 1 до n
    array[i]=0
    повторити
    для j від 1 до m
        якщо matrix[i][j]>=0
            то
                array[i]+=matrix[i][j]
        все якщо
    все повторити
все повторити
```

Все підпрограма

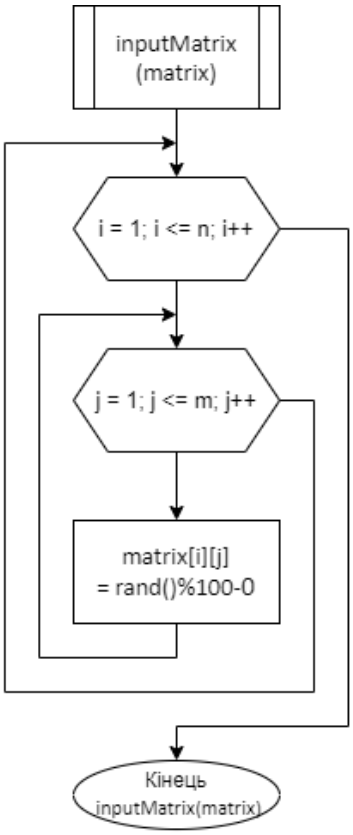
```
Підпрограма sortArray(array)
    повторити
    для і від 2 до n
        t=array[i]
        повторити
        для j від i-1 до 1 та array[j]>t з кроком -1
            array[j+1]=array[j]
            array[j]=t
        все повторити
    все повторити
Все підпрограма
```

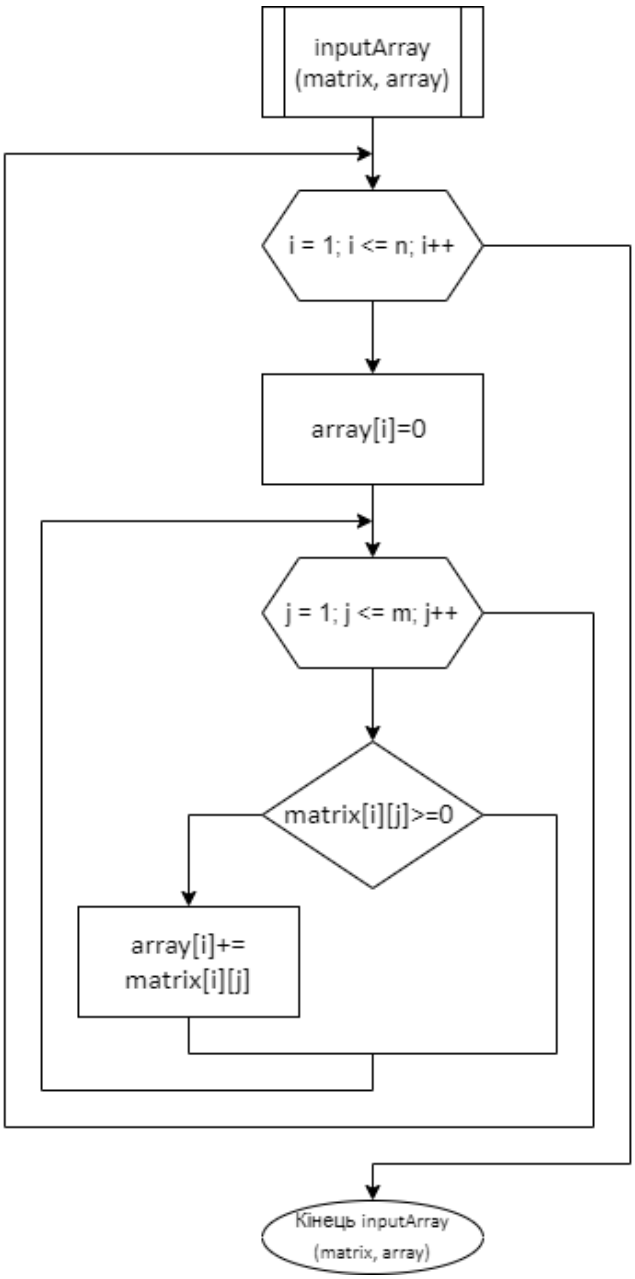
Блок-схема алгоритму

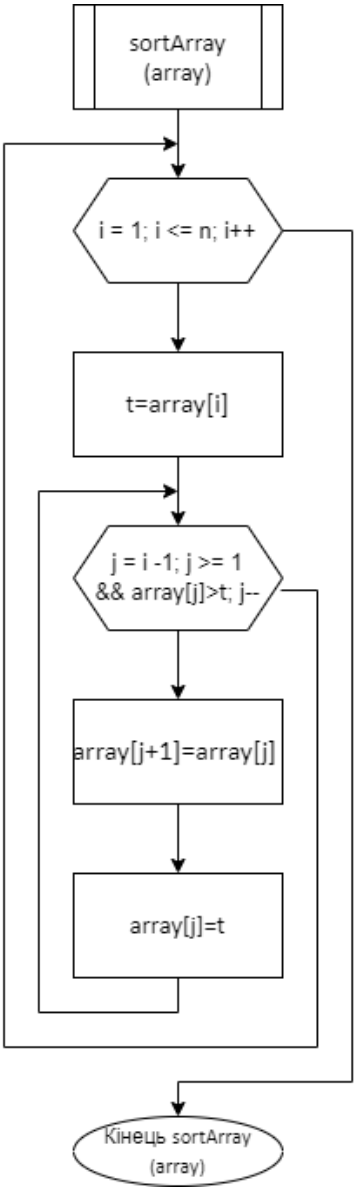



Початок
n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix
(matrix)
inputArray
(matrix, array)
outputMatrix
(matrix)
Виведення
одновимірного
масиву
Сортування
одновимірного
масиву
Виведення
відсортованого
одновимірного
масиву
Кінець
Початок
n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix
(matrix)
inputArray
(matrix, array)
outputMatrix
(matrix)
outputArray
(array)
Сортування
одновимірного
масиву
Виведення
відсортованого
одновимірного
масиву
Кінець
Початок
n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix
(matrix)
inputArray
(matrix, array)
outputMatrix
(matrix)
outputArray
(array)
sortArray(array)
Виведення
відсортованого
одновимірного
масиву
Кінець
Початок
n = 6
m = 5
matrix[n][m]
array[n]
inputMatrix
(matrix)
inputArray
(matrix, array)
outputMatrix
(matrix)
outputArray
(array)
sortArray(array)
outputArray
(array)
Кінець

x


inputMatrix
(matrix)
i = 1; i <= n; i++
j = 1; j <= m; j++
matrix[i][j]
= rand()%100-0
Кінець
inputMatrix(matrix)


inputArray
(matrix, array)
i = 1; i <= n; i++
array[i]=0
j = 1; j <= m; j++
matrix[i][j]>=0
array[i]+=
matrix[i][j]
Кінець inputArray
(matrix, array)


sortArray
(array)
i = 1; i <= n; i++
t=array[i]
j = i -1; j >= 1
&& array[j]>t; j--
array[j+1]=array[j]
array[j]=t
Кінець sortArray
(array)

Код програми

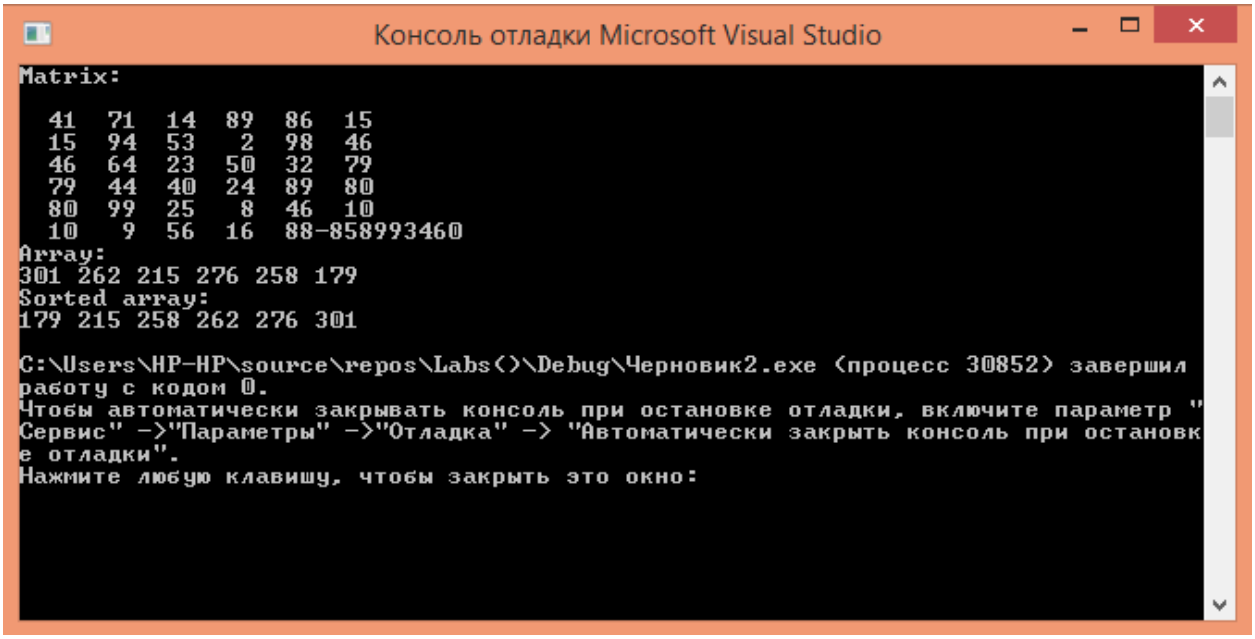
```cpp
#include <iostream>
#include <iomanip>
using namespace std;

const int n = 6;
const int m = 5;

void inputMatrix(int[][m]);
void inputArray(int[][m], int[]);
void sortArray(int[]);
void outputMatrix(int[][m]);
void outputArray(int[]);

int main()
{
    srand(time(NULL));
    int matrix[n][m];
    int array[n];
    inputMatrix(matrix);
    inputArray(matrix, array);
    cout << "Matrix: \n";
    outputMatrix(matrix);
    cout << "\n";
    cout << "Array: \n";
    outputArray(array);
    sortArray(array);
    cout << "Sorted array: \n";
    outputArray(array);
}
```

```cpp
void inputMatrix(int matrix[][m])
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            matrix[i][j] = rand() % 100 - 0;
        }
    }
}

void inputArray(int matrix[][m], int array[])
{
    for (int i = 0; i < n; i++)
    {
        array[i] = 0;
        for (int j = 0; j < m; j++)
        {
            if (matrix[i][j] >= 0)
            {
                array[i] += matrix[i][j];
            }
        }
    }
}
```

```
void sortArray(int array[])
{
    int t;
    for (int i = 1; i < n; i++)
    {
        t = array[i];
        for (int j = i - 1; j >= 0 && array[j] > t; j--)
        {
            array[j + 1] = array[j];
            array[j] = t;
        }
    }
}

void outputMatrix(int matrix[][m])
{
    for (int i = 0; i < n; i++)
    {
        cout << "\n";
        for (int j = 0; j < n; j++)
        {
            cout << setw(4) << matrix[i][j];
        }
    }
}
```

```
void outputArray(int array[])
{
    for (int i = 0; i < n; i++)
    {
        cout << array[i] << " ";
    }
    cout << "\n";
}
```

Випробування коду



Висновок. У результаті лабораторної роботи було досліджено алгоритми пошуку та сортування, набуто практичні навички використання цих алгоритмів під час складання програмних специфікацій. Було поставлено задачу, побудовано математичну модель, розроблено алгоритм її вирішення у вигляді псевдокоду, який було переведено на блок-схему. Алгоритм усмішно генерує двовимірний масив та одновимірний масив, елементами якого є суми додатних елементів рядків першого масиву та сортує його за зростанням використовуючи метод вставки.