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

«Дослідження арифметичних циклічних алгоритмів»
Варіант 17

Виконав студент ІП-13 Козак Антон Миколайович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вечерковська Анастасія Сергіївна
(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

№ варіанта	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
17	6 x 5	Цілий	Із суми додатних значень елементів рядків двовимірного масиву. Відсортувати методом вставки за зростанням.

Постановка задачі

Заданий алгоритм повинен:

1. Генерувати двовимірний масив цілих чисел із заданою розмірністю.
2. Ініціювати одновимірний масив цілих чисел, кожен елемент якого є сумою додатних значень елементів відповідних рядків двовимірного масиву.
3. Відсортувати одновимірний масив методом вставки за зростанням

Побудова математичної моделі

Таблиця змінних

Змінна	Тип	Ім’я	Призначення
Кількість рядків двовимірного масиву	Цілий	n	Початкові дані
Кількість стовпців двовимірного масиву	Цілий	m	Початкові дані
Двовимірний масив цілих чисел	Цілий	matrix	Проміжні дані
Тимчасов змінна	Цілий	temp	
Одновимірний масив цілих чисел	Цілий	array	Кінцеві дані

Використані функції

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

Власні функції

- generateMatrix(двовимірний цілочисельний масив, ціле число, ціле число) – повертає цілочисельний двовимірний масив заданої розмірності, кожен елемент якого згенерований випатковим чином.
- generateArray(двовимірний цілочисельний масив, одновимірний цілочисельний масив, ціле число, ціле число) – повертає цілочисельний одновимірний масив заданої розмірності, кожен елемент якого є сумою додатних елементів відповідних рядків двовимірного масиву.
- sortArray(одновимірний цілочисельний масив, ціле число) – повертає цілочисельний одновимірний масив заданої розмірності, елементи якого відсортовані за зростанням.

1. Згенеруємо перший масив за допомогою функції rand: matrix[i][j] = rand()%200-100.
2. Визначимо кожен поточний елемент одновимірного масиву нулем. За наявності додатних чисел у відповідному рядку двовимірного масиву, додамо їм до поточного елементу.
3. Відсортуємо одновимірний масив методом вставок.

Розв’язання

1. Визначимо основні дії.
2. Деталізуємо дію оголошення масивів.
3. Деталізуємо дію генерації двовимірного масиву.
4. Деталізуємо дію генерації одновимірного масиву.
5. Деталізуємо дію сортування одновимірного масиву.
6. Деталізуємо дію генерації елементів двовимірного масиву випадковим чином за допомогою ітераційної форми повторення.
7. Деталізуємо дію генерації елементів одновимірного масиву сумою додатних значень за допомогою ітераційної форми повторення та умовної форми вибору.
8. Деталізуємо дію сортування одновимірного масиву за зростанням за допомогою ітераційної форми повторення.

Псевдокод алгоритму

Крок 1
Початок
    Введення n, m
    Оголошення масивів
    Генерації двовимірного масиву
    Генерації одновимірного масиву
    Сортування одновимірного масиву
    Виведення array
Кінець

Підпрограма
    generateMatrix(matrix, n, m)
    Генерації елементів двовимірного масиву випадковим чином
Все підпрограма

Підпрограма
    generateArray(matrix, array, n, m)
    Генерації елементів одновимірного масиву сумою додатних значень
Все підпрограма

Підпрограма
    sortArray(array, n)
    Сортування одновимірного масиву за зростанням
Все підпрограма

Крок 2

Початок

Введення n, m
matrix[n][m]
array[n]
Генерації двовимірного масиву
Генерації одновимірного масиву
Сортування одновимірного масиву
Виведення array

Кінець

Підпрограма

generateMatrix(matrix, n, m)
Генерації елементів двовимірного масиву випадковим чином

Все підпрограма

Підпрограма

generateArray(matrix, array, n, m)
Генерації елементів одновимірного масиву сумою додатних значень

Все підпрограма

Підпрограма

sortArray(array, n)
Сортування одновимірного масиву за зростанням

Все підпрограма

Крок 3

Початок

Введення n, m
matrix[n][m]
array[n]
generateMatrix(matrix, n, m)
Генерації одновимірного масиву
Сортування одновимірного масиву
Виведення array

Кінець

Підпрограма

generateMatrix(matrix, n, m)
Генерації елементів двовимірного масиву випадковим чином

Все підпрограма

Підпрограма

generateArray(matrix, array, n, m)
Генерації елементів одновимірного масиву сумою додатних значень

Все підпрограма

Підпрограма

sortArray(array, n)
Сортування одновимірного масиву за зростанням

Все підпрограма

Крок 4

Початок

Введення n, m
matrix[n][m]
array[n]
generateMatrix(matrix, n, m)
generateArray(matrix, array, n, m)
Сортування одновимірного масиву
Виведення array

Кінець

Підпрограма

generateMatrix(matrix, n, m)
Генерації елементів двовимірного масиву випадковим чином

Все підпрограма

Підпрограма

generateArray(matrix, array, n, m)
Генерації елементів одновимірного масиву сумою додатних значень

Все підпрограма

Підпрограма

sortArray(array, n)
Сортування одновимірного масиву за зростанням

Все підпрограма

Крок 5

Початок

Введення n, m
matrix[n][m]
array[n]
generateMatrix(matrix, n, m)
generateArray(matrix, array, n, m)
sortArray(array, n)
Виведення array

Кінець

Підпрограма

generateMatrix(matrix, n, m)
Генерації елементів двовимірного масиву випадковим чином

Все підпрограма

Підпрограма

generateArray(matrix, array, n, m)
Генерації елементів одновимірного масиву сумою додатних значень

Все підпрограма

Підпрограма

sortArray(array, n)
Сортування одновимірного масиву за зростанням

Все підпрограма

Крок 6

Початок

Введення n, m
matrix[n][m]
array[n]
generateMatrix(matrix, n, m)
generateArray(matrix, array, n, m)
sortArray(array, n)
Виведення array

Кінець

Підпрограма

generateMatrix(matrix, n, m)
повторити
для i **від** 1 **до** n
повторити
для j **від** 1 **до** m
matrix[i][j] = rand()%200-100
все повторити
все повторити

Все підпрограма

Підпрограма

generateArray(matrix, array, n, m)
Генерації елементів одновимірного масиву сумою додатних значень

Все підпрограма

Підпрограма

sortArray(array, n)
Сортування одновимірного масиву за зростанням

Все підпрограма

Крок 7

Початок

Введення n, m
matrix[n][m]
array[n]
generateMatrix(matrix, n, m)
generateArray(matrix, array, n, m)
sortArray(array, n)
Виведення array

Кінець

Підпрограма

generateMatrix(matrix, n, m)
повторити
для i **від** 1 **до** n
повторити
для j **від** 1 **до** m
matrix[i][j] = rand()%200-100
все повторити
все повторити

Все підпрограма

Підпрограма

generateArray(matrix, array, n, m)
повторити
для i **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, n)
Сортування одновимірного масиву за зростанням

Все підпрограма

Крок 8

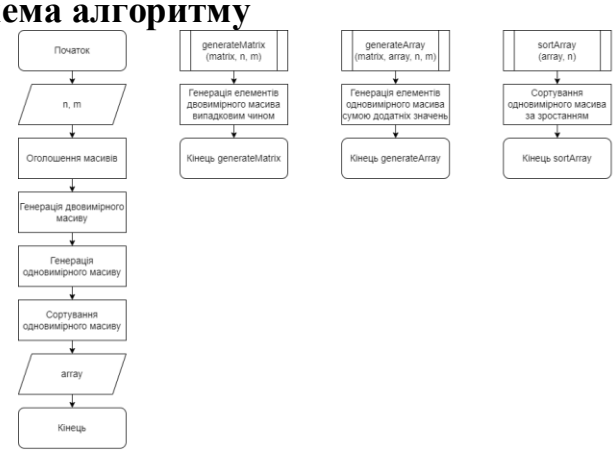
Початок
    Введення n, m
    matrix[n][m]
    array[n]
    generateMatrix(matrix, n, m)
    generateArray(matrix, array, n, m)
    sortArray(array, n)
    Виведення array
Кінець

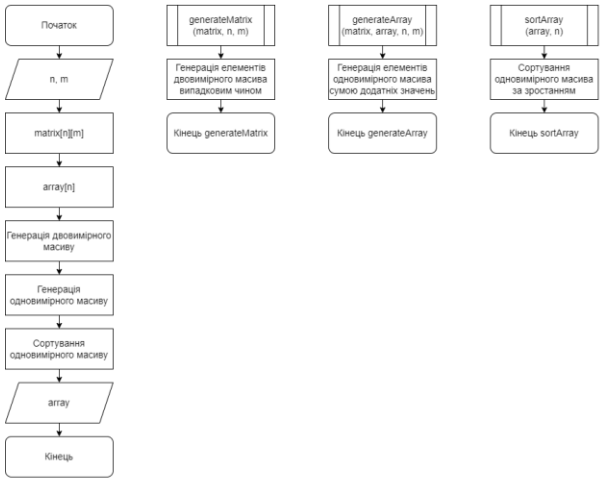
Підпрограма
    generateMatrix(matrix, n, m)
    повторити
    для i **від** 1 **до** n
        повторити
        для j **від** 1 **до** m
            matrix[i][j] = rand()%200-100
        все повторити
    все повторити
Все підпрограма

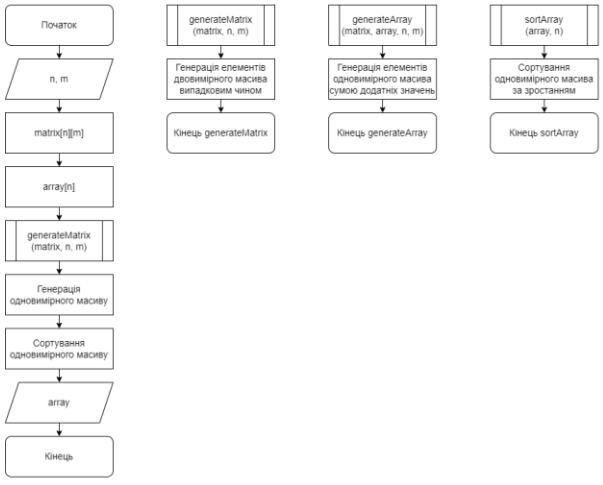
Підпрограма
    generateArray(matrix, array, n, m)
    повторити
    для i **від** 1 **до** n
        array[i]=0
        повторити
        для j **від** 1 **до** m
            якщо matrix[i][j]>0
                то
                    array[i]+=matrix[i][j]
            все якщо
        все повторити
    все повторити
Все підпрограма

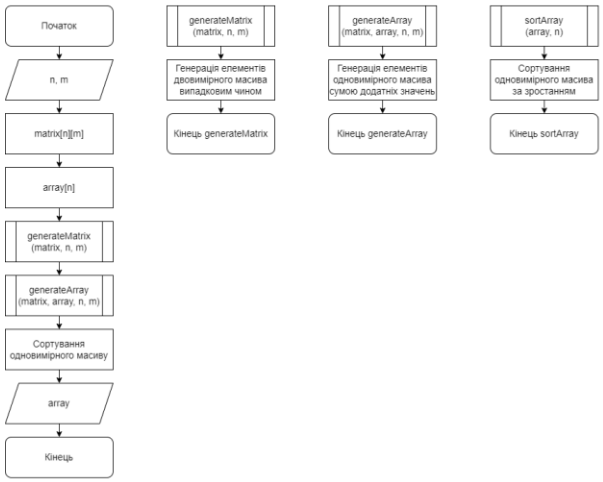
```
Підпрограма
    sortArray(array, n)
    повторити
    для i від 2 до n
        temp=array[i]
        повторити
        для j від i-1 до 1 та array[j]>temp з кроком -1
            array[j+1]=array[j]
            array[j]=temp
        все повторити
    все повторити
Все підпрограма
```

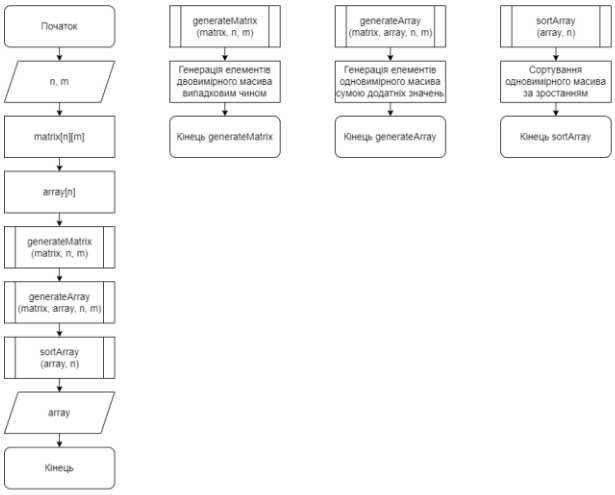
Блок-схема алгоритму

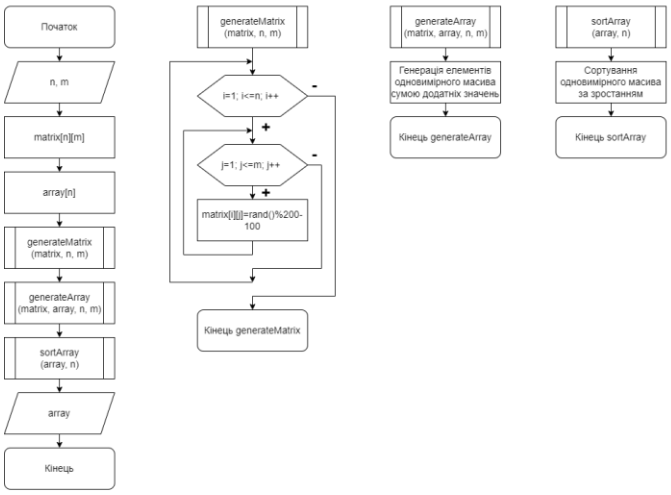


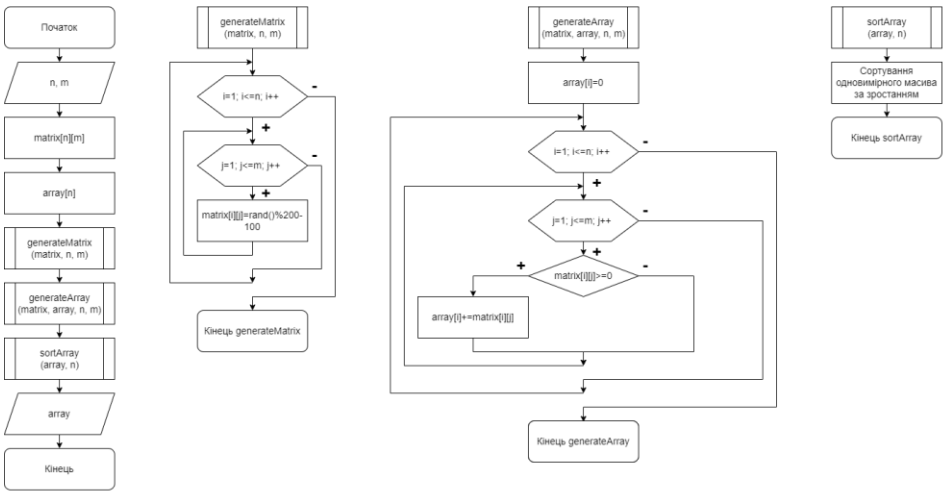


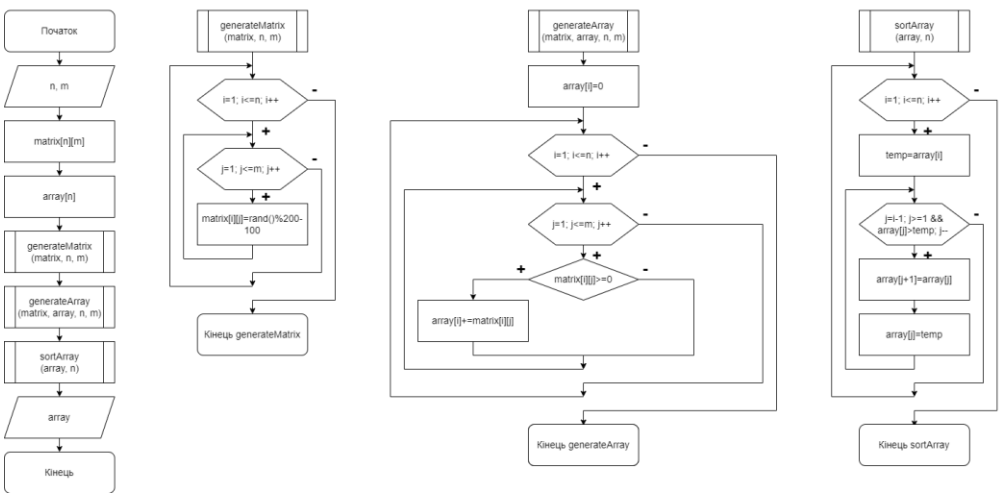





Початок
n, m
matrix[n][m]
array[n]
generateMatrix (matrix, n, m)
generateArray (matrix, array, n, m)
sortArray (array, n)
array
Кінець
generateMatrix (matrix, n, m)
Генерація елементів двовимірного масива випадковим чином
Кінець generateMatrix
generateArray (matrix, array, n, m)
Генерація елементів одновимірного масива сумою додатніх значень
Кінець generateArray
sortArray (array, n)
Сортування одновимірного масива за зростанням
Кінець sortArray

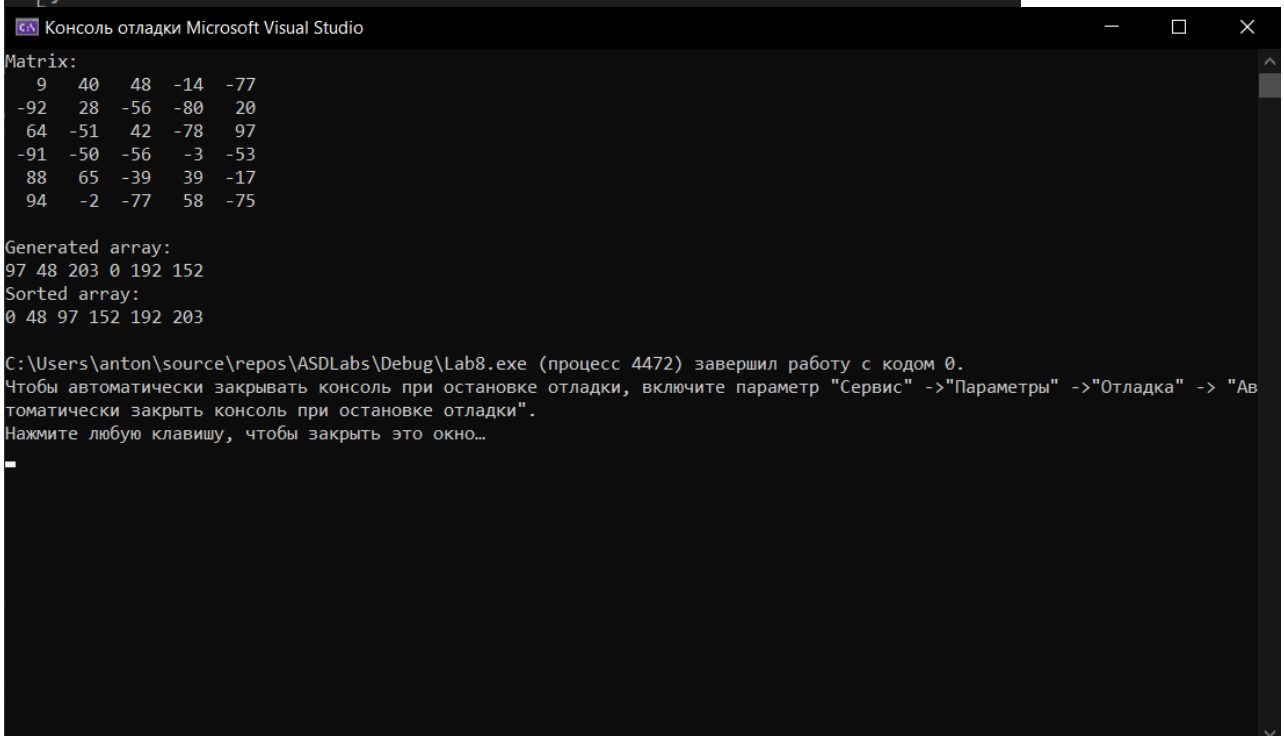

Початок
n, m
matrix[n][m]
array[n]
generateMatrix (matrix, n, m)
generateArray (matrix, array, n, m)
sortArray (array, n)
array
Кінець
generateMatrix (matrix, n, m)
i=1; i<=n; i++
j=1; j<=m; j++
matrix[i][j]=rand()%200-100
Кінець generateMatrix
generateArray (matrix, array, n, m)
Генерація елементів одновимірного масива сумою додатніх значень
Кінець generateArray
sortArray (array, n)
Сортування одновимірного масива за зростанням
Кінець sortArray


Початок
n, m
matrix[n][m]
array[n]
generateMatrix (matrix, n, m)
generateArray (matrix, array, n, m)
sortArray (array, n)
array
Кінець
generateMatrix (matrix, n, m)
i=1; i<=n; i++
j=1; j<=m; j++
matrix[i][j]=rand()%200-100
Кінець generateMatrix
generateArray (matrix, array, n, m)
array[i]=0
matrix[i][j]>=0
array[i]+=matrix[i][j]
Кінець generateArray
sortArray (array, n)
Сортування одновимірного масива за зростанням
Кінець sortArray


Початок
n, m
matrix[n][m]
array[n]
generateMatrix (matrix, n, m)
generateArray (matrix, array, n, m)
sortArray (array, n)
array
Кінець
generateMatrix (matrix, n, m)
i=1; i<=n; i++
j=1; j<=m; j++
matrix[i][j]=rand()%200-100
Кінець generateMatrix
generateArray (matrix, array, n, m)
array[i]=0
matrix[i][j]>=0
array[i]+=matrix[i][j]
Кінець generateArray
sortArray (array, n)
temp=array[i]
j=i-1; j>=1 && array[j]>temp; j--
array[j+1]=array[j]
array[j]=temp
Кінець sortArray

```cpp
/* Козак Антон ІП-13
Варіант 17
Двовимірний масив: цілий, 6x5. Одновимірний масив: утворений із суми додатніх елементів рядків двовимірного масиву.
Відсортувати одновимірний масив методом вставки за зростанням.*/

#include <iostream>
using namespace std;
const int n = 6, m = 5; //розмірність масивів

void generateMatrix(int[][m]); //генерує двовимірний масив
void generateArray(int[][m], int[]); //генерує одновимірний масив
void sortArray(int[]); //сортує одновимірний масив за зростанням
void outputMatrix(int[][m]); //
void outputArray(int[]); //

int main()
{
    srand(time(NULL));
    int matrix[n][m], array[n]; // двовимірний та одновимірний масив
    //генерація масивів
    generateMatrix(matrix);
    generateArray(matrix, array);
    //виведення масивів
    cout << "Matrix:\n";
    outputMatrix(matrix);
    cout << "Generated array:\n";
    outputArray(array);
    //сортування та виведення одновимірного масиву
    sortArray(array);
    cout << "Sorted array: \n";
    outputArray(array);

}
void generateMatrix(int matrix[][m])
{
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            matrix[i][j] = rand() % 200 - 100;
        }
    }
}

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

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

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



Випробування алгоритму

Блок	Дія
	Початок
1.	**Введення** n=6, m=5
2.	generateMatrix(matrix, 6, 5)
3.	i=1; i<=6; i++
4.	j=1;j<=5; j++
5.	matrix[1][1]=rand()&200-100 = 9
6.	j=2;j<=5; j++
7.	matrix[1][2]=rand()&200-100 = 40
8.	j=3;j<=5; j++
9.	matrix[1][3]=rand()&200-100 = 48
10.	j=4;j<=5; j++
11.	matrix[1][4]=rand()&200-100 = -14
12.	j=5;j<=5; j++
13.	matrix[1][5]=rand()&200-100 = -77
14.	j=6;j>5
15.	i=2; i<=6; i++
16.	j=1;j<=5; j++
17.	matrix[2][1]=rand()&200-100 = -92
18.	j=2;j<=5; j++
19.	matrix[2][2]=rand()&200-100 = 28
20.	j=3;j<=5; j++
21.	matrix[2][3]=rand()&200-100 = -56
22.	j=4;j<=5; j++
23.	matrix[2][4]=rand()&200-100 = -80
24.	j=5;j<=5; j++
25.	matrix[2][5]=rand()&200-100 = 20
26.	j=6;j>5
27.	i=3; i<=6; i++
28.	j=1;j<=5; j++
29.	matrix[3][1]=rand()&200-100 =64
30.	j=2;j<=5; j++
31.	matrix[3][2]=rand()&200-100 = -51
32.	j=3;j<=5; j++
33.	matrix[3][3]=rand()&200-100 = 42
34.	j=4;j<=5; j++
35.	matrix[3][4]=rand()&200-100 = -78
36.	j=5;j<=5; j++
37.	matrix[3][5]=rand()&200-100 = 97
38.	j=6;j>5
39.	i=4; i<=6; i++
40.	j=1;j<=5; j++
41.	matrix[4][1]=rand()&200-100 = -91
42.	j=2;j<=5; j++

43.	matrix[4][2]=rand()&200-100 = -50
44.	j=3;j<=5; j++
45.	matrix[4][3]=rand()&200-100 = -56
46.	j=4;j<=5; j++
47.	matrix[4][4]=rand()&200-100 = -3
48.	j=5;j<=5; j++
49.	matrix[4][5]=rand()&200-100 = -53
50.	j=6;j>5
51.	i=5; i<=6; i++
52.	j=1;j<=5; j++
53.	matrix[5][1]=rand()&200-100 = 88
54.	j=2;j<=5; j++
55.	matrix[5][2]=rand()&200-100 = 65
56.	j=3;j<=5; j++
57.	matrix[5][3]=rand()&200-100 = -39
58.	j=4;j<=5; j++
59.	matrix[5][4]=rand()&200-100 = 39
60.	j=5;j<=5; j++
61.	matrix[5][5]=rand()&200-100 = -17
62.	j=6;j>5
63.	i=6; i<=6; i++
64.	j=1;j<=5; j++
65.	matrix[6][1]=rand()&200-100 = 94
66.	j=2;j<=5; j++
67.	matrix[6][2]=rand()&200-100 = -2
68.	j=3;j<=5; j++
69.	matrix[6][3]=rand()&200-100 = -77
70.	j=4;j<=5; j++
71.	matrix[6][4]=rand()&200-100 = 58
72.	j=5;j<=5; j++
73.	matrix[6][5]=rand()&200-100 = -75
74.	j=6;j>5
75.	i=7;i>6
76.	generateArray(matrix, array, n, m)
77.	i=1; i<=6; i++
78.	array[1]=0
79.	j=1;j<=5; j++
80.	9>=0
81.	array[1]=9
82.	j=2;j<=5; j++
83.	40>=0
84.	array[1]=49
85.	j=3;j<=5; j++
86.	48>=0
87.	array[1]=97

88.	j=4;j<=5; j++
89.	-14<0
90.	j=5;j<=5; j++
91.	-77<0
92.	j=6;j>5
93.	i=2; i<=6; i++
94.	array[2]=0
95.	j=1;j<=5; j++
96.	-92<0
97.	j=2;j<=5; j++
98.	28>=0
99.	array[1]=28
100.	j=3;j<=5; j++
101.	-56>=0
102.	j=4;j<=5; j++
103.	-80<0
104.	j=5;j<=5; j++
105.	20>=0
106.	array[2]=48
107.	j=6;j>5
108.	i=3; i<=6; i++
109.	array[3]=0
110.	j=1;j<=5; j++
111.	64>=0
112.	array[3]=64
113.	j=2;j<=5; j++
114.	-51<0
115.	j=3;j<=5; j++
116.	42>=0
117.	array[3]=106
118.	j=4;j<=5; j++
119.	-78<0
120.	j=5;j<=5; j++
121.	97>=0
122.	array[3]=203
123.	j=6;j>5
124.	i=4; i<=6; i++
125.	array[4]=0
126.	j=1;j<=5; j++
127.	-91<0
128.	j=2;j<=5; j++
129.	-50<0
130.	j=3;j<=5; j++
131.	-56<0
132.	j=4;j<=5; j++

133.	-3<0
134.	j=5;j<=5; j++
135.	-53<0
136.	j=6;j>5
137.	i=5; i<=6; i++
138.	array[5]=0
139.	j=1;j<=5; j++
140.	88>=0
141.	array[5]=88
142.	j=2;j<=5; j++
143.	65>=0
144.	array[5]=153
145.	j=3;j<=5; j++
146.	-39<0
147.	j=4;j<=5; j++
148.	39>=0
149.	array[5]=192
150.	j=5;j<=5; j++
151.	-17<0
152.	j=6;j>5
153.	i=6; i<=6; i++
154.	array[6]=0
155.	j=1;j<=5; j++
156.	94>=0
157.	array[6]=94
158.	j=2;j<=5; j++
159.	-2<0
160.	j=3;j<=5; j++
161.	-39<0
162.	j=4;j<=5; j++
163.	58>=0
164.	array[6]=152
165.	j=5;j<=5; j++
166.	-75<0
167.	j=6;j>5
168.	i=7;i>6
169.	sortArray(array, n)
170.	i=2; i<=6; i++
171.	temp = 48
172.	j=1;j>=1, 97>temp
173.	array[2]=97
174.	array[1]=48
175.	j=0; j<1
176.	i=3; i<=6; i++
177.	temp = 203

178.	j=2;j>=1, 97<temp
179.	i=4; i<=6; i++
180.	temp = 0
181.	j=3;j>=1, 203>temp;j--
182.	array[4]=203
183.	array[3]=0
184.	j=2;j>=1, 97>temp;j--
185.	array[3]=97
186.	array[2]=0
187.	j=1;j>=1, 48>temp;j--
188.	array[2]=48
189.	array[1]=0
190.	j=0;j<1
191.	i=5; i<=6; i++
192.	temp = 192
193.	j=4;j>=1, 203>temp;j--
194.	array[5]=203
195.	array[4]=192
196.	j=3;j>=1, 97<temp
197.	i=6; i<=6; i++
198.	temp = 152
199.	j=5;j>=1, 203>temp;j--
200.	array[6]=203
201.	array[5]=152
202.	j=4;j>=1, 192>temp;j--
203.	array[5]=192
204.	array[4]=152
205.	j=3;j>=1, 97<temp
206.	i=7; i>6
207.	**Виведення** array = {0, 48, 97, 152, 192, 203}
	Кінець

Висновки

Протягом виконання цієї лабораторної роботи я набув навичок використання алгоритмів пошуку та сортування під час складання програмних специфікацій. Маючи розмірність та тип масиву, я склав алгоритм, який усмішно генерує заданий масив, одновимірний масив, елементами якого є суми додатних елементів рядків першого масиву та сортує його за зростанням використовуючи метод вставки.