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 алгоритмів пошуку та сортування»Варіант

№20

Виконав студент ІП-12 Логвиненко Владислав Олексiйович

(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 8**

Дослідження алгоритмів пошуку та сортування

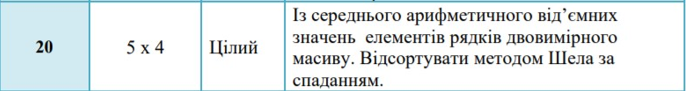
Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Варіант 20**

**Задача.**

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

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом.
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.



**Постановка задачі**

Знайти значення елементів одновимірного массиву із середнього арифметичного від’ємних значень елементів рядків двовимірного масиву. Відсортувати методом Шела за спаданням. Ініціюємо двовимірний масив випадковими значеннями в проміжку:[-10;10]. Далі знаходимо середнє арифметичного від’ємних елементів кожного рядка і ініціюємо одновимірний масив цими значеннями. За допомогою метода Шела сортуємо цей масив за спаданням. Результатом розв’язку буде відсортований за спаданням одновимірний масив середніх арифметичних від’ємних значень початкового двовимірного масиву.

Побудова математичної моделі.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Двовимірний масив | Цілий | arr | Проміжні дані |
| Одновимірний масив | Цілий | arr\_1 | Результат |
| Розмір n | Цілий | n | Початкове дані |
| Розмір m | Цілий | m | Початкове дані |
| Сума | Цілий | sum | Проміжні дані |
| Кількість від’ємних елементів у рядку | Цілий | k | Проміжні дані |
| Змінна temp | Дійсний | temp | Проміжні дані |
| Крок | Цілий | d | Проміжні дані |
| Діапазон значень елементів масиву | Цілий | RANGE | Початкове дане |
| Лічильник i | Порядковий | i | Проміжне дане |
| Лічильник j | Порядковий | j | Проміжне дане |

Таким чином математичне формулювання задачі зводиться до:

!) створеннядвовимірного масиву 5x4 з випадкових цілими числами у проміжку від -10 до 10(Використовую функцію rand() – за допомогою, якої заповняю масив випадковими значеннями).

2)пошуку середнього арифметичного від’ємних елементів в кожному рядку даного двовимірного масиву (використовуватиму лінійний пошук).

3) Кожне знайдене значення буде елементом одновимірного масиву.

4)формування одновимірного масиву необхідно його відсортувати за спаданням. Метод сортування за умовою – сортування Шела

## 2) Розв’язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію опису констант

*Крок 3.* Деталізуємо дію опсу та ініціювання двовимірного масиву.

*Крок 4.* Деталізуємо дію опису та ініціювання одновимірного масиву.

*Крок 5.* Деталізуємо дію сортування одновимірного масиву.

*Крок 6.* Деталізуємо дію виводу значень елементів масивів.

3) Псевдокод

Крок 1.

**Початок**

Деталізуємо дію опису констант

Деталізуємо дію опсу та ініціювання двовимірного масиву.

Деталізуємо дію опису та ініціювання одновимірного масиву.

Деталізуємо дію сортування одновимірного масиву.

Деталізуємо дію виводу значень елементів масивів.

**кінець**

Крок 2.

**початок**

n = 5,

m = 4,

arr[n][m],

arr\_1[n]

Деталізуємо дію опсу та ініціювання двовимірного масиву.

Деталізуємо дію опису та ініціювання одновимірного масиву.

Деталізуємо дію сортування одновимірного масиву.

Деталізуємо дію виводу значень елементів масивів.

**кінець**

Крок 3.

**початок**

n = 5,

m = 4,

arr[n][m],

arr\_1[n]

**initial(arr[][], n, m):**

**output(arr[][], n, m):**

Деталізуємо дію опису та ініціювання одновимірного масиву.

Деталізуємо дію сортування одновимірного масиву.

Деталізуємо дію виводу значень елементів масивів.

Крок 4.

n = 5,

m = 4,

arr[n][m],

arr\_1[n]

**initial(arr[][], n, m):**

**output(arr[][], n, m):**

**average(arr[][], n, m):** k=0, sum=0, arr\_1[5]

**print(arr\_1[], n):**

Деталізуємо дію сортування одновимірного масиву.

Деталізуємо дію виводу значень елементів масивів.

**кінець**

Крок 5.

**початок**

n = 5,

m = 4,

arr[n][m],

arr\_1[n]

**initial(arr[][], n, m):**

**output(arr[][], n, m):**

**average(arr[][], n, m):** k=0, sum=0, arr\_1[5]

**print(arr\_1[], n):**

**Shell(arr\_1[], n):**

Деталізуємо дію виводу значень елементів масивів.

**кінець**

Крок 6.

**початок**

n = 5,

m = 4,

arr[n][m],

arr\_1[n]

**initial(arr[][], n, m):**

**output(arr[][], n, m):**

**average(arr[][], n, m):** k=0, sum=0, arr\_1[5]

**print(arr\_1[], n):**

**Shell(arr\_1[], n):**

**print(arr\_1[], n):**

**кінець**

*Псевдокод(підпрогрма -* **initial(arr[][], n, m)***):*

**initial(arr[][], n, m):**

**повторити для і від 0 до n не включно зкроком1:**

**повторити для j від 0 до m не включно з кроком 1:**

arr[i][j]=rand() % 21 - 10

**все повторити**

**все повторити**

**кінець**

*Псевдокод(підпрогрма -* **output(arr[][], n, m):***):*

**output(arr[][], n, m):**

**повторити для і від 0 до n не включно зкроком1:**

**повторити для j від 0 до m не включно з кроком 1:**

print arr\_1[i]

**все повторити**

**все повторити**

*Псевдокод(підпрогрма -* **Shell(arr\_1[], n):***):*

temp, d = n/2

**поки d>0**

**повторити для і від 0 до (n-d) не включно з кроком 1:** j = i

**поки** ( j>=0 && arr\_1[j]< arr\_1[j+d])

temp= arr\_1[j]

arr\_1[j] = arr\_1[j+d]

arr\_1[j+d] = temp

j--

все повтории

все повтории

*Псевдокод(підпрогрма -* **average(arr\_1[], n):***):k=0 , sum = 0 ,arr\_1[5]*

***Повторити для і від 0 до n не включно з кроком 1:***

***Повторити для j від 0 до n не включно з кроком 1:***

***Якщо (arr[i][j]<0)***

***То***

***Sum += arr[i][j]***

***K++***

***Все якщо***

***Якщо(k == 0)***

***То***

***Arr\_1[i] =0***

***Інакше***

***Arr\_1[i] =sum/k***

***Все якщо***

k=0

sum=0

***все повторити***

***все повторити***

*Псевдокод(підпрогрма -* **print(arr\_1[], n):**

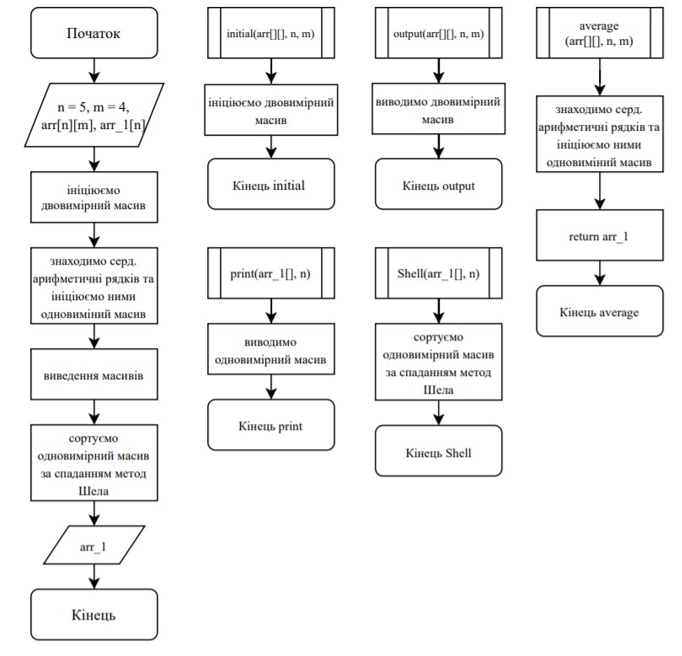
**print(arr\_1[], n):**

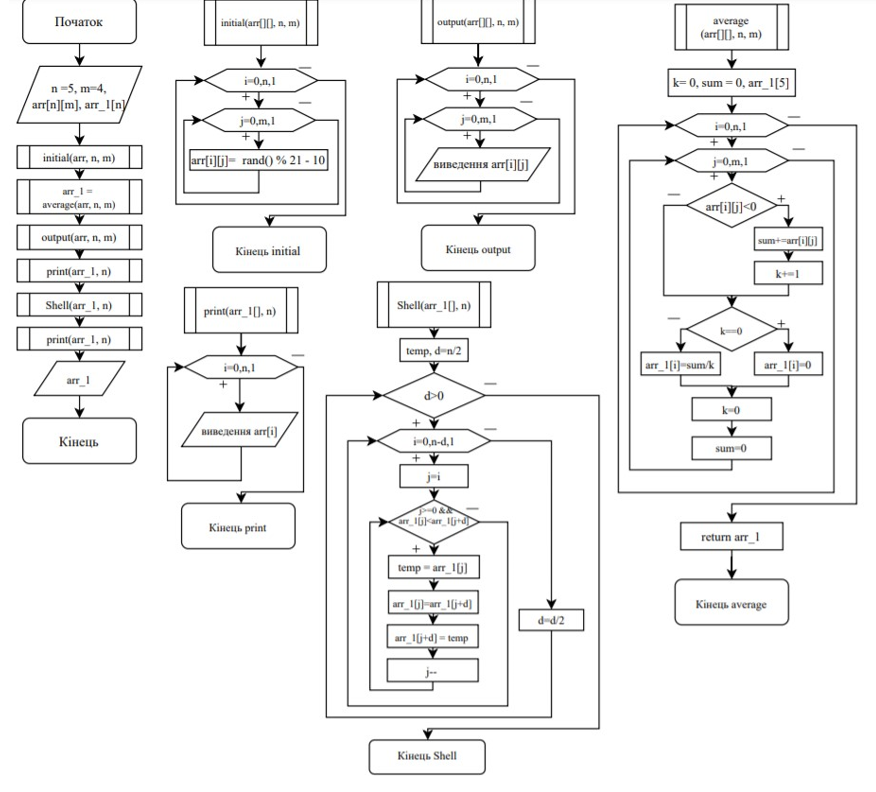
**повторити для і від 0 до n не включно з кроком 1:**

виведення arr\_1[i]

**все повторити**

**Все процедура**

***4) Блок-схема***

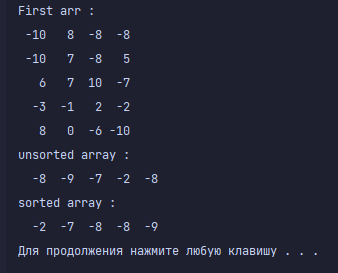


***Код програми на мові С++***

#include <iostream>  
#include <ctime>  
#include <iomanip>  
#define **n** 5  
#define **m** 4  
*using namespace* std;  
*void* initial(*int* arr\_1[**n**][**m**]);  
*void* output(*int* arr\_1[**n**][**m**],string massage);  
*int* \*average(*int* arr\_1[**n**][**m**]);  
*void* print(*int* arr\_2[**m**],string massage);  
*void* Shell(*int* arr\_2[**n**]);  
*int* main() {  
 *int* arr\_1[**n**][**m**],\*arr\_2;  
 initial(arr\_1);  
 output(arr\_1,"First arr :");  
 arr\_2 = average(arr\_1);  
 print(arr\_2,"unsorted array :");  
 Shell(arr\_2);  
 print(arr\_2,"sorted array :");  
 system("pause");  
 *return* 0;  
}  
*void* initial(*int* arr\_1[**n**][**m**]){  
 srand(time(**NULL**));  
 *for* (*int* i=0;i<**n**;i++){  
 *for* (*int* j=0;j<**m**;j++){  
 arr\_1[i][j] = rand() % 21 - 10;  
 }  
 }  
}  
*void* output(*int* arr\_1[**n**][**m**],string massage){  
 cout << massage << endl;  
 *for* (*int* i=0;i<**n**;i++){  
 *for* (*int* j=0;j<**m**;j++){  
 cout<< setw(4) << arr\_1[i][j];  
 }  
 cout << endl;  
 }  
}  
*int* \*average(*int* arr\_1[**n**][**m**]){  
 *int* k = 0,sum = 0;  
 *int* \*arr\_2 = *new int* [**n**];  
 *for* (*int* i=0;i<**n**;i++){  
 *for* (*int* j=0;j<**m**;j++){  
 *if* (arr\_1[i][j] < 0){  
 sum += arr\_1[i][j];  
 k++;  
 }  
 }  
 *if*(k == 0)  
 arr\_2[i] = 0;  
 *else* arr\_2[i] = sum/k;  
 k = 0;  
 sum = 0;  
 }  
 *return* arr\_2;  
}  
*void* print(*int* arr\_2[**n**],string massage){  
 cout << massage << endl;  
 *for* (*int* i=0;i<**n**;i++){  
 cout << setw(4) << arr\_2[i];  
 }  
 cout << endl;  
}  
*void* Shell(*int* arr\_2[**n**]){  
 *double* temp;  
 *int* d = **n** / 2;  
 *while* (d > 0){  
 *for*(*int* i = 0; i < **n** - d;i++) {  
 *int* j = i;  
 *while*(j >= 0 && arr\_2 [j] < arr\_2[j + d]){  
 temp = arr\_2[j];  
 arr\_2[j] = arr\_2[j + d];  
 arr\_2[j + d] = temp;  
 j--;  
 }  
 }  
 d = d / 2;  
 }  
}

}

**Віпробування кода:**



**5) Висновок -** Ми дослідили методи алгоритми пошуку та сортування та набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій підпрограм. В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження впорядкованого за спаданням одновимірного масиву, який складається з середніх арифметичних значень рядків початкового двовимірного масиву за допомогою підпрограм(функцій та процедур) та з врахуванням умов, визначили основні дії.