

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 34

Виконав студент Щербацький Антон ПІ-14
(шифр, прізвище, ім'я, по батькові)

Перевірив Доцент кафедри іпІ Мартинова О. П.
(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 - 34

Постановка задачі:

34	5 x 7	Дійсний	Із середнього арифметичного додатних значень елементів стовпців двовимірного масиву. Відсортувати обміном за спаданням.
----	-------	---------	---

Для створення та виведення двовимірного масиву вивкористаємо арифметичний цикл for. Для заповнення елементів випадковими числами застосуємо функцію rand() у діапазоні від -100 до 100. За допомогою вкладених арифметичних циклів та оператора умови знайдемо середні значення суми додатніх елементів кожного стовпця масиву. Створимо додаткову змінну counter для знаходження кількості додатніх елементів. Також створимо змінну temp для збереження проміжних елементів сортування. Для сортування обміном буде застосована конструкція із двох вкладених арифметичних циклів та оператора подання умови.

Математична модель:

Змінна	Тип	Ім`я	Призначення
Двовимірний масив	Індексований(натуральний)	Arr1[ROWS][COLS]	Проміжні дані
Масив середніх значень	Індексований(натуральний)	Arr2[COLS]	Результат
Рядки масиву	Натуральний	ROWS	Проміжні дані
Стовпці масиву	Натуральний	COLS	Проміжні дані
Середні значення сум масиву	Дійсний	average	Проміжні дані

Тимчасова змінна для збереження елементу сортування	Дійсний	temp	Проміжні дані
Сума елементів стовпців	Дійсний	sum	Проміжні дані

Крок 1. Визначимо основні дії

Крок 2. Ініціалізація змінних ROWS, COLS, arr1[ROWS][COLS], arr2[COLS], counter, sum, average, temp.

Крок 3. Деталізуємо заповнення та виведення двовимірного масиву.

Крок 4. Деталізуємо знаходження масиву із середніми значеннями суми стовпців.

Крок 5. Деталізуємо сортування обміном масиву із середніми значеннями.

Крок 6. Виведення відсортованого масиву.

Псевдокод

Крок 1

Початок

Ініціалізація ROWS, COLS, arr1[ROWS][COLS], arr2[COLS], counter, sum, average, temp.

Заповнення та виведення двовимірного масиву.

Заповнення масиву із середніми значеннями

Сортування обміном масиву із середніми значеннями

Виведення відсортованого масиву

Кінець

Крок 2

Початок

ROWS = 5, COLS = 7, arr1[ROWS][COLS], arr2[COLS], counter, sum, average, temp

Заповнення та виведення двовимірного масиву.

Заповнення масиву із середніми значеннями

Сортування обміном масиву із середніми значеннями

Виведення відсортованого масиву

Кінець

Крок 3

Початок

ROWS = 5, COLS = 7, arr1[ROWS][COLS], arr2[COLS], counter, sum, average,
temp

Повторити

Для I від 0 до 5, $I = I + 1$

Повторити

Для j від 0 до 7, $j = j + 1$

Arr1[i][j] = rand від -100 до 100

Повторити

Все повторити

Заповнення масиву із середніми значеннями

Сортування обміном масиву із середніми значеннями

Виведення відсортованого масиву

Кінець

Крок 4

Початок

ROWS = 5, COLS = 7, arr1[ROWS][COLS], arr2[COLS], counter, sum, average,
temp

Повторити

Для I від 0 до 5, $I = I + 1$

Повторити

Для j від 0 до 7, $j = j + 1$

Arr1[i][j] = rand від -100 до 100

Все повторити

Все повторити

Повторити

Для I від 0 до 7, $I = i + 1$

Sum = 0

Counter = 0

Повторити

Для I від 0 до 5, $i = I + 1$

Якщо arr1[i][j] >= 0

То

Sum = sum + arr1[i][j]

Counter = counter + 1

Все якщо

Все повторити

Average = sum/counter

Якщо counter = 0

То

Arr2[i] = 0

Інакше

Arr2[i] = average

Все якщо

Все повторити

Сортування обміном масиву із середніми значеннями

Виведення відсортованого масиву

Кінець

Крок 5

Початок

ROWS = 5, COLS = 7, arr1[ROWS][COLS], arr2[COLS], counter, sum, average, temp

Повторити

Для I від 0 до 5, I = I + 1

Повторити

Для j від 0 до 7, j = j + 1

Arr1[i][j] = rand від -100 до 100

Все повторити

Все повторити

Повторити

Для I від 0 до 7, I = i + 1

Sum = 0

Counter = 0

Повторити

Для I від 0 до 5, i = I + 1

Якщо arr1[i][j] >= 0

То

Sum = sum + arr1[i][j]

Counter = counter + 1

Все якщо

Все повторити

Average = sum/counter

Якщо counter = 0

То

Arr2[i] = 0

Інакше

Arr2[i] = average

Все якщо

Все повторити

Повторити

Для I від 0 до 6, I = I + 1

Повторити

Для j від 0 до 6, j = j + 1

Якщо arr2[j + 1] < arr2[j]

То

Temp = arr2[j + 1]

Arr2[j + 1] = arr2[j]

Arr2[j] = temp

Все якщо

Все повторити

Все повторити

Виведення відсортованого масиву

Кінець

Крок 6

Початок

ROWS = 5, COLS = 7, arr1[ROWS][COLS], arr2[COLS], counter, sum, average, temp

Повторити

Для I від 0 до 5, I = I + 1

Повторити

Для j від 0 до 7, j = j + 1

Arr1[i][j] = rand від -100 до 100

Все повторити

Все повторити

Повторити

Для I від 0 до 7, I = i + 1

Sum = 0

Counter = 0

Повторити

Для I від 0 до 5, i = I + 1

Якщо arr1[i][j] >= 0

То

Sum = sum + arr1[i][j]

Counter = counter + 1

Все якщо

Все повторити

Average = sum/counter

Якщо counter = 0

То

Arr2[i] = 0

Інакше

Arr2[i] = average

Все якщо

Все повторити

Повторити

Для I від 0 до 6, I = I + 1

Повторити

Для j від 0 до 6, j = j + 1

Якщо arr2[j + 1] < arr2[j]

То

Temp = arr2[j + 1]

Arr2[j + 1] = arr2[j]

Arr2[j] = temp

Все якщо

Все повторити

Все повторити

Повторити

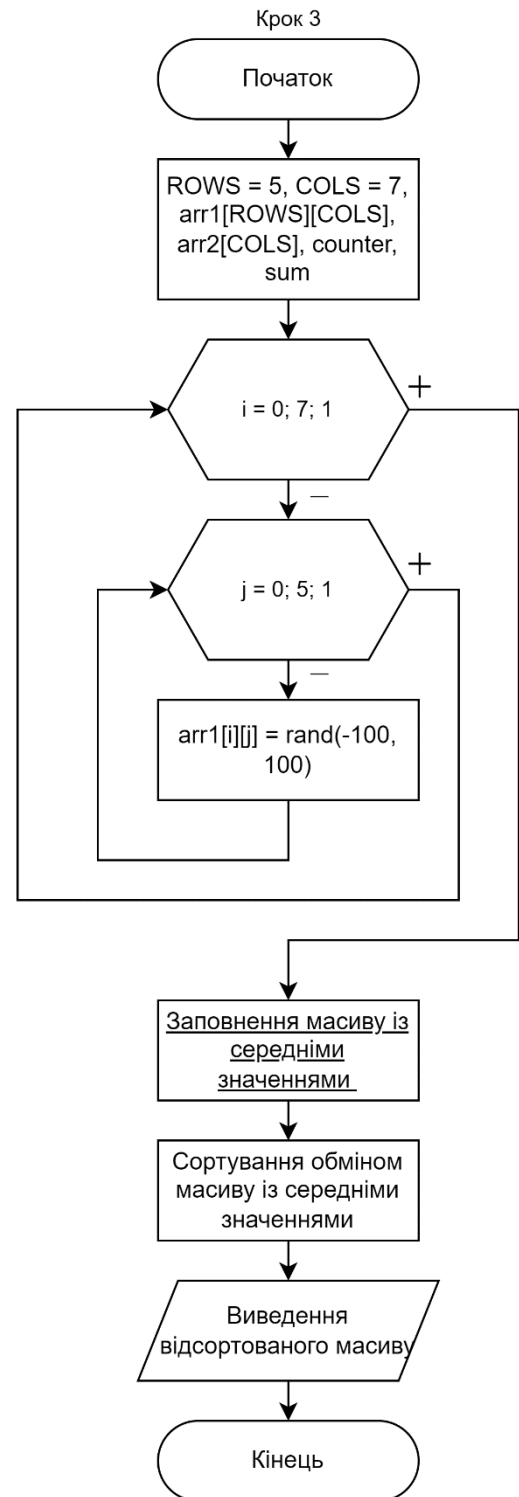
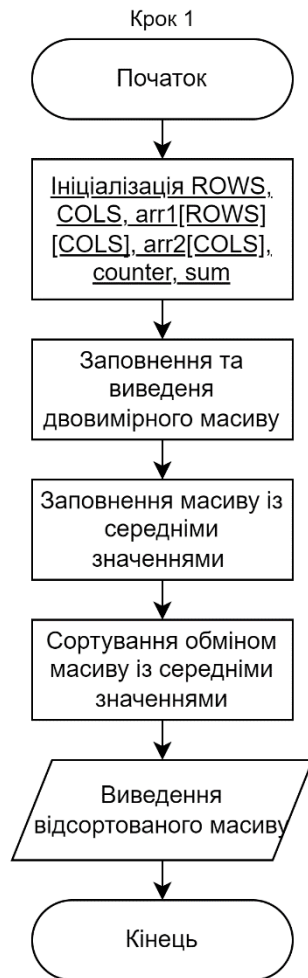
Для I від 0 до 7, I = I + 1

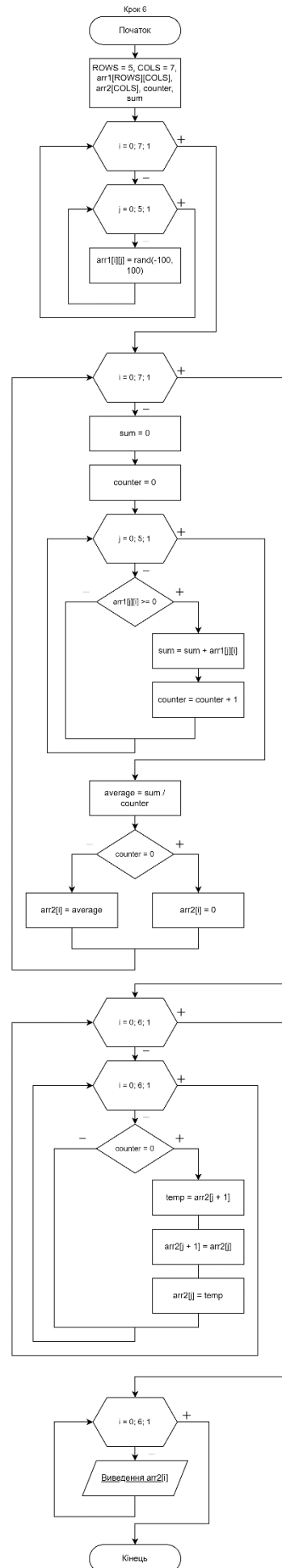
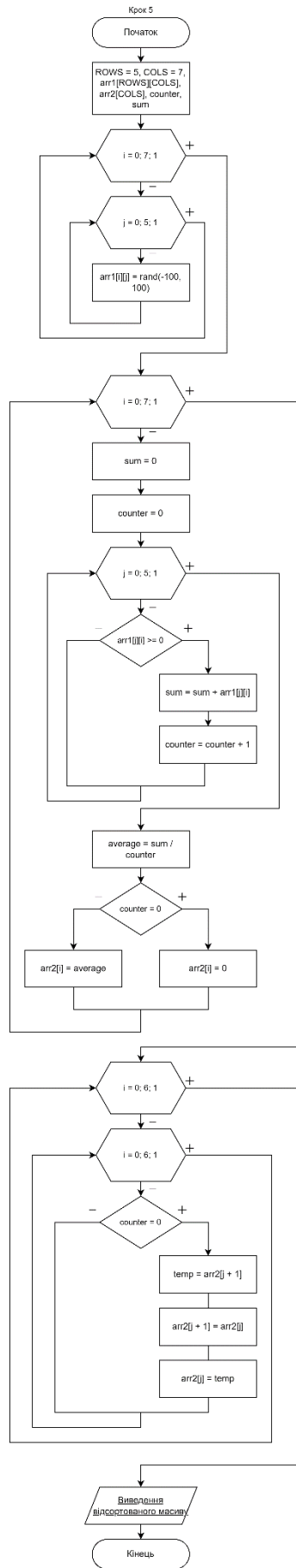
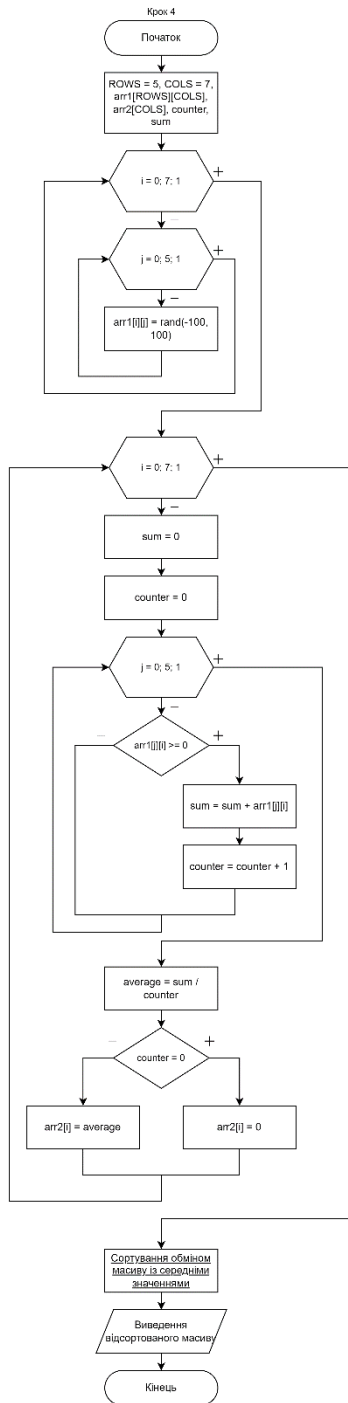
Виведення arr2[i]

Все повторити

Кінець

Блок-схема





Код програми(C++)

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      srand(_Seed: time(_Time: NULL));
8      const int ROWS = 5, COLS = 7;
9      int arr1[ROWS][COLS]{}; counter, sum;
10     float arr2[COLS]{}; average, temp;
11
12     cout << endl;
13
14     for(int i = 0; i < ROWS; i++)
15     {
16         for(int j = 0; j < COLS; j++)
17         {
18             arr1[i][j] = rand() % 201 - 100;
19             cout << setw(4) << arr1[i][j];
20         }
21         cout << endl;
22     }
23
24     cout << endl;
25     cout << "Average sum without sort:" << '\t';
26
27     for(int i = 0; i < COLS; i++)
28     {
29         sum = 0;
30         counter = 0;
```

```

31     for(int j = 0; j < ROWS; j++)
32     {
33         if(arr1[j][i] >= 0)
34         {
35             sum += arr1[j][i];
36             counter++;
37         }
38     }
39     average = (float)sum/counter;
40     if(counter == 0)
41         arr2[i] = 0;
42     else
43         arr2[i] = average;
44
45     cout << fixed << setprecision(1) << arr2[i] << " ";
46 }
47
48 cout << endl;
49 cout << "Average sum with sort: " << '\t';
50
51 for(int i = 0; i < 6; i++)
52 {
53     for(int j = 0; j < 6; j++)
54     {
55         if(arr2[j + 1] < arr2[j])
56         {
57             temp = arr2[j + 1];
58             arr2[j + 1] = arr2[j];
59             arr2[j] = temp;
60         }
61     }
62 }
63
64 for(int i = 0; i < 7; i++)
65 {
66     cout << arr2[i] << " ";
67 }
68 cout << endl;
69 }

```

```

26 -93  78 -69 -56  31  93
12  23 -63  85  57  26   8
32  16  41  31  45  74  12
48  85 -85 -43  57  -9 -38
-8  53 100 -29 -13 -50 -69

```

```

Average sum without sort:      29.5  44.2  73.0  58.0  53.0  43.7  37.7
Average sum with sort:        29.5  37.7  43.7  44.2  53.0  58.0  73.0

```

```

Process finished with exit code 0

```

Тестування програми

Крок	Дія
1	$Arr2[1] = 26 + 12 + 32 + 48 = 118 / 4 = 29.5$
2	$Arr2[2] = 23 + 16 + 85 + 53 = 177 / 4 = 44.2$
3	$Arr2[3] = 78 + 41 + 100 = 219 / 3 = 73$

4	$Arr2[4]=85+31=116/2=58$
5	$Arr2[5]=57+45+57=159/3=53$
6	$Arr2[6]=31+26+74=131/3=43.7$
7	$Arr2[7]=93+8+12=113/3=37.7$
8	$Arr2[i]=29.5\ 37.7\ 43.7\ 44.2\ 53\ 58$

Висновок

На цій лабораторній роботі було досліджено подання алгоритмів пошуку та сортування, набуто навичок використання цих алгоритмів під час виконання даного варіанту роботи. Було побудовано математичну модель, блок-схему, псевдокод та випробування алгоритму. На прикладі мого варіанту за допомогою вкладених арифметичних циклів та оператора умови було знайдемо середні значення суми додатніх елементів кожного стовпця масиву та відсортовано методом обміну.