

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 2

Виконав студент ПІ-15, Богун Даниїл Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірила 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 2

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 .

№ варіанта	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
2	6 x 5	Дійсний	Із середнього арифметичного значення елементів рядків двовимірного масиву. Відсортувати обміном за спаданням.

Постановка задачі

Створюємо двовимірний масив, розміром 6 x 5 і заповнюємо його. Потім створюємо підпрограми, які будуть рахувати суму елементів кожного рядка, середнє арифметичне значення цих елементів, створюємо одновимірний масив із цими значеннями і сортуємо його.

Змінна	Тип	Ім'я	Призначення
Матриця	Дійсний	<i>arr</i>	Результат, проміжні дані
Одновимірний масив	Дійсний	<i>arr1</i>	Результат, проміжні дані
Кількість рядків матриці	Цілий	<i>R</i>	Початкове дане
Кількість стовпців матриці	Цілий	<i>ST</i>	Початкове дане
Лічильник	Цілий	<i>counter</i>	Проміжне дане
Лічильник циклу	Цілий	<i>i</i>	Параметр циклу
Лічильник вкладеного циклу	Цілий	<i>j</i>	Параметр циклу
Сума елементів ряду	Дійсний	<i>Sum</i>	Проміжне дане
Генерація матриці	Процедура	<i>matrix</i>	Початкове дане
Сер. арифметичне елементів ряду	Дійсний	<i>sr</i>	Проміжне дане

Змінна для впорядкування елементів	Дійсний	tmp	Проміжне дане
Знаходження суми елементів ряду	Процедура	SumRowEl	Проміжне дане
Знаходження сер. арифм. елементів	Процедура	SerArifm	Результат
Створення одновимірного масиву	Процедура	srArr	Проміжне дане
Сортування масиву	Процедура	SortObmin	Результат

Спочатку в окремій підпрограмі `matrix` створюємо матрицю 6 x 5 та заповнюємо її випадковими дійсними числами від -10 до 10. Потім створюємо функцію `SumRowEl`, в якій за допомогою циклу рахуємо суму елементів кожного рядка. Суму позначаємо як `Sum`. `Sum` повертаємо.

В підпрограмі `SerArifm` рахуємо середнє арифметичне значень елементів кожного рядка. Позначаємо як `sr`. `sr` повертаємо.

В підпрограмі `srArr` створюємо одновимірний масив і заповнюємо його значеннями `sr`.

В підпрограмі `SortObmin` сортуємо цей одновимірний масив бульбашкою за спаданням.

Псевдокод алгоритму:

Основна програма:

Початок

$R = 6$

$ST = 5$

Counter = 0

Генерація випадкових чисел

matrix(arr, R, ST)

Повторити

Для i від 0 до R

Sum = **SumRowEl**(arr, R, ST, i)

counter = counter + 1

Вивід «Сума елементів» , counter, «рядка = », Sum

SerArifm(arr, R, ST, i)

arr1[i] = **srArr**(arr, R, ST, i)

Все повторити

SortObmin(arr1, R)

Кінець

Підпрограми:

matrix(arr, R, ST)

Повторити

Для i від 0 до R

Повторити

Для j від 0 до ST

$m[i][j]$ = рандомне число
від -10 до 10

Вивід $m[i][j]$

Все повторити

Все повторити

Кінець

SerArifm(arr, R, ST, i)

Sum = **SumRowEl**(arr, R, ST, i)

sr = Sum / ST

Повернути sr

SumRowEl(arr, R, ST, i)

Sum = 0

Повторити

Для j від 0 до ST

Sum = Sum $m[i][j]$

Все повторити

Повернути Sum

srArr(arr, R, ST, i)

arr1 = **SerArifm**(arr, R, ST, i)

Повернути arr1

SortObmin(arr1, R)

Для i від 0 до R

Повторити

Для j від 0 до ST

tmp = arr[j+1]

arr[j+1] = arr[j]

arr[j] = tmp

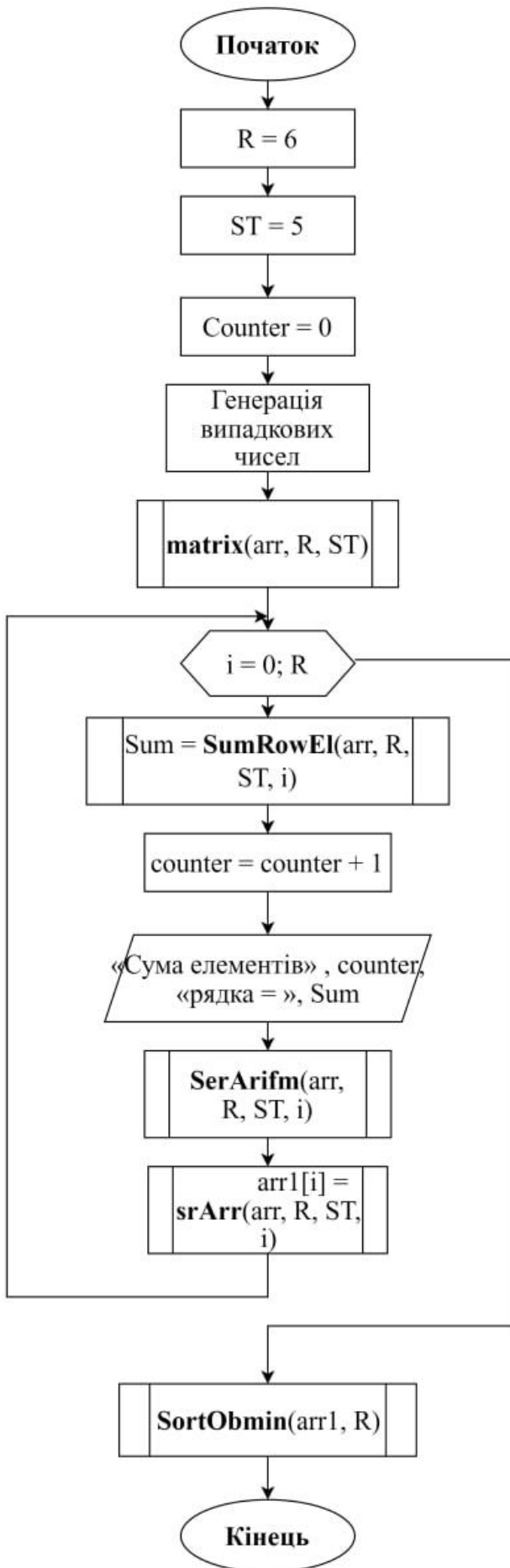
Все повторити

Вивід arr[j]

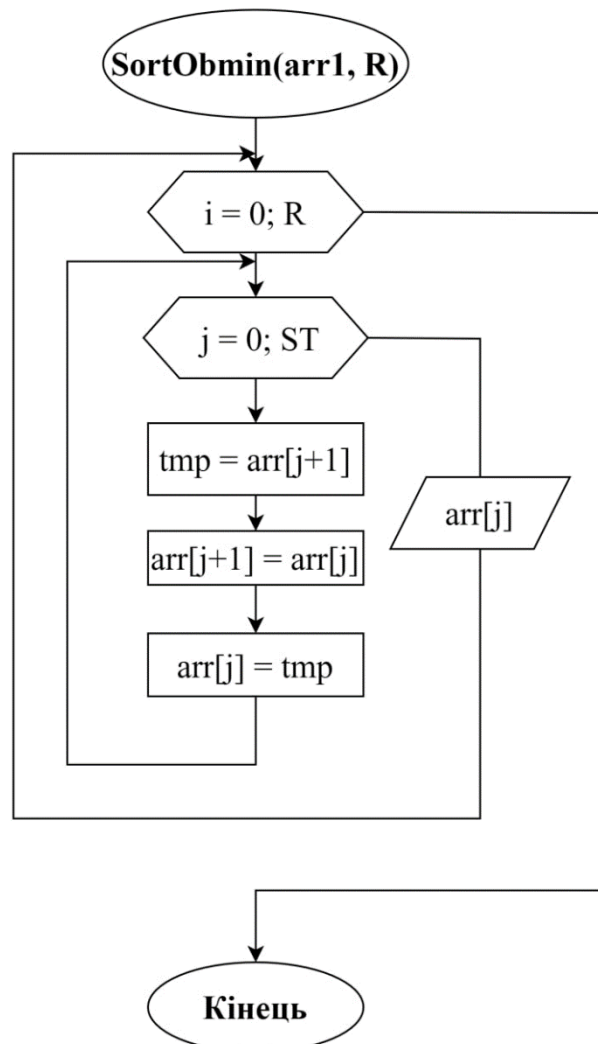
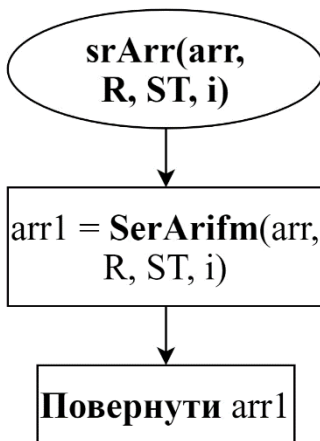
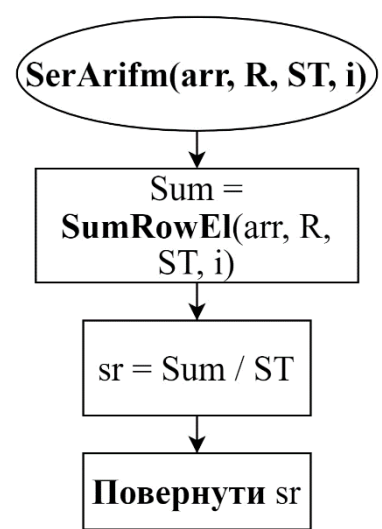
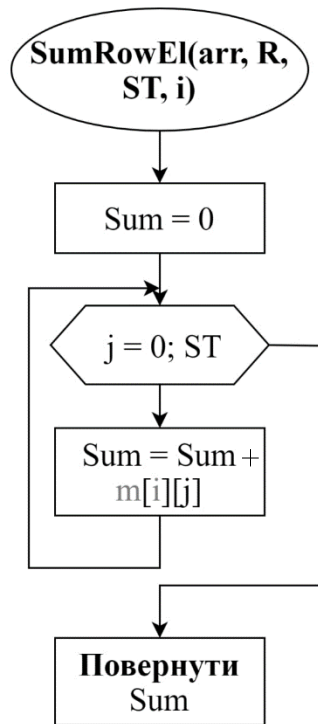
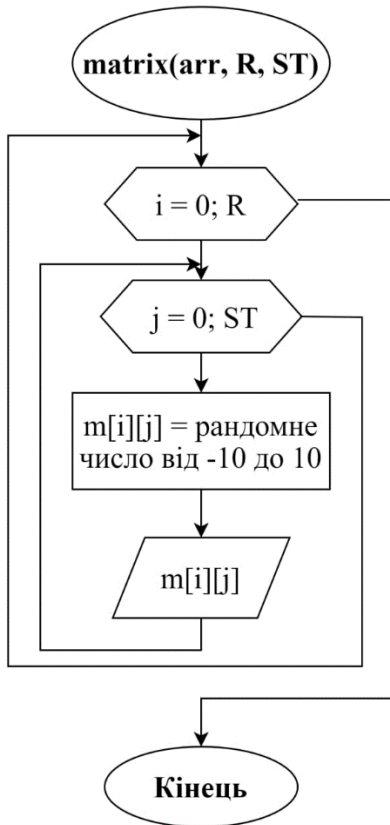
Все повторити

Кінець

Основна програма:



Підпрограми:



Код:

```

#include <iomanip>
#include <iostream>
#include <windows.h>
#include <ctime>
void matrix(double m[6][5], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            m[i][j] = ((double)rand()) / RAND_MAX * 20 - 1 + 1 + (-10);
            std::cout << std::setw(12)<< m[i][j];
        }
        std::cout << "\n";
    }
}

```

```

double SumRowEl(double m[6][5], int row, int col, int i) {
    double Sum = 0;
    for (int j = 0; j < col; j++) {
        Sum += m[i][j];
    }
    return Sum;
}

```

```

double SerArifm(double m[6][5], int row, int col, int i) {
    double Sum = SumRowEl(m, row, col, i);
    double sr = Sum / col;
    return sr;
}

```

```

double srArr(double m[6][5], int row, int col, int i) {
    double* m1 = &m[0][0];
    m1[i] = SerArifm(m, row, col, i);
    return m1[i];
}

```

```

void SortObmin(double arr[], int row) {
    int i, j;
    double tmp;
    for (i = 0; i < row; i++) {
        for (j = 0; j < row - i - 1; j++)
            if (arr[j] > arr[j + 1]) {
                tmp = arr[j + 1];
                arr[j + 1] = arr[j];
                arr[j] = tmp;
            }
        std::cout << arr[j] << std::setw(14);
    }
}

```

```

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int R = 6;
    const int ST = 5;
    int counter = 0;
    srand(time(NULL));
    double arr[R][ST];
    std::cout << "Двомірний масив 6 x 5: " << "\n\n";
    matrix(arr, R, ST);
    std::cout << "\n";
}

```



```

for (int i = 0; i < R; i++) {
    double Sum = SumRowEl(arr, R, ST, i);
    counter++;
    std::cout << "\n" << " Сума елементів " << counter << " рядка = " << Sum;
    SerArifm(arr, R, ST, i);
}
std::cout << "\n\n";
double arr1[R];

std::cout << "\n\n" << "Результат :" << "\n";
for (int i = 0; i < R; i++) {
    arr1[i] = srArr(arr, R, ST, i);
}
SortObmin(arr1, R);
std::cout << "\n";
}

```

Випробування:

Консоль отладки Microsoft Visual Studio

Двумірний масив 6 x 5:

9.3762	7.116	-3.32987	-9.30601	-2.45888
9.29563	9.99145	-8.4991	-8.24335	-0.0955229
8.90133	-8.96542	-1.06601	-2.67739	-9.07285
8.71639	-5.1915	2.36183	-0.419019	6.1681
8.034	-1.51646	6.20655	-5.82629	1.81677
-0.600909	9.67895	6.07776	-3.78826	-5.32273

Сума елементів 1 рядка = 1.39744
 Сума елементів 2 рядка = 2.44911
 Сума елементів 3 рядка = -12.8803
 Сума елементів 4 рядка = 11.6358
 Сума елементів 5 рядка = 8.71456
 Сума елементів 6 рядка = 6.0448

Результат :

2.32716	1.74291	1.20896	0.489822	0.279489	-2.57607
---------	---------	---------	----------	----------	----------

Консоль отладки Microsoft Visual Studio

Двовірний масив 6 x 5:

9.45799	-3.91217	3.71075	-7.13553	-4.34309
7.55974	7.87835	-1.75207	4.9498	6.05335
5.20005	-7.86309	-7.05374	-8.61507	6.27308
2.59133	5.91357	3.44279	5.39354	-5.14145
-6.29994	5.88488	1.94372	9.71007	7.24784
-3.86151	2.91238	-1.62816	6.2804	4.80209

Сума елементів 1 рядка = -2.22205
Сума елементів 2 рядка = 24.6892
Сума елементів 3 рядка = -12.0588
Сума елементів 4 рядка = 12.1998
Сума елементів 5 рядка = 18.4866
Сума елементів 6 рядка = 8.5052

Результат :

4.93783	3.69732	2.43995	1.70104	-0.444411	-2.41176
---------	---------	---------	---------	-----------	----------

Консоль отладки Microsoft Visual Studio

Двовірний масив 6 x 5:

9.52574	-0.861538	-5.57299	-7.77459	9.70519
9.04721	-7.53227	-2.49916	7.59758	-1.53172
2.1305	-6.94876	-7.62871	-5.24644	6.8041
2.39113	2.43873	5.80187	3.38481	3.04178
-9.40611	7.14469	-6.46962	-4.72396	-9.71252
5.62975	5.10666	0.274972	-6.34693	7.83197

Сума елементів 1 рядка = 5.02182
Сума елементів 2 рядка = 5.08164
Сума елементів 3 рядка = -10.8893
Сума елементів 4 рядка = 17.0583
Сума елементів 5 рядка = -23.1675
Сума елементів 6 рядка = 12.4964

Результат :

3.41166	2.49928	1.01633	1.00436	-2.17786	-4.6335
---------	---------	---------	---------	----------	---------