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 №9 з дисципліни

«Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження алгоритмів обходу масивів»

Варіант 31

Виконав: студент ІП-15 Ткач Владислав Анатолійович

Перевірив:_____

Київ 2021

Лабораторна робота 9

Дослідження алгоритмів обходу масивів

Мета - дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Завдання

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. Обчислення змінної, що описана в п.1, згідно з варіантом

Постановка задачі

Результатом розв’язку є двомірний масив у якому змінено місцями мінімальний елемент та елемент над головною діагональю, який більший за середнє арифметичне.

Побудова математичної моделі

Складемо таблицю імен змінних:

Змінна	Тип	Ім’я	Призначення
К-сть рядків масиву	дійсне	n	Початкове дане
К-сть стовбців масиву	дійсне	m	Початкове дане
Двомірний масив	дійсне	A	Початкове дане
Мінімальний елемент	дійсне	X	Проміжне дане
Індекс елементу по рядках	ціле	index_x	проміжне дане
Індекс елементу по стовбцях	ціле	index_y	проміжне дане
Середнє	дійчне	arg	проміжне дане
Значення масиву (запам’ятовування)	дійсне	znach	проміжне дане
Сума елментів	дійсне	sum	проміжне дане
Напрямок руху	ціле	dir	проміжне дане
Знаходження мінімального	Функція	find_X()	Обрахунок даних
Знаходження середнього	Функція	find_arg()	Обрахунок даних
Заміна елементів	Функція	replace()	Обрахунок даних
Ініціювання масиву	Функція	mas_in()	Ввід даних
Вивід двомірного масиву	Функція	mas_out()	Вивід даних

Випадкове значення	функція	rand()	Обрахунок даних
Зміна типу	функція	double()	Обрахунок даних
Максимальне випадкове знасчення	ціле	RAND_MAX	Проміжне дане
Максимальне значення	Дісне	DBL_MAX	Проміжне дане

Для визначенн результату зробимо наступні дії:

Ініціалізуємо двомірний масив випадковими значеннями використовуючи функцію **mas_in(),** у якій використовуємо формулу для визначення випадкового дійсного числа від 0 до 10: **(double)(rand()) / RAND_MAX * 10.**

Знайдемо мінімальний елемент використовуючи функцію **find_X(),**, в якій використаємо обхід по стовбчиках. Функція знаходить індекс мінімального елементу.

Знайдемо середнє арифметичне використовуючи функцію **find_arg(),** в якій використаємо обхід по стовбчиках. Функція знаходить індекс мінімального елементу.

Змінимо місця найменший елемент із елементом над головною діагоналлю, який більший за середнє.

Для зручності також використаємо функції виведення масиву:

mas_out() – виведення двомірного масиву

Розв’язання

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію ініціювання двомірного масиву

Крок 3. Деталізуємо дію визначення мінімального елементу та його індекс

Крок 4. Деталізуємо дію визначення середнього арифметичного.

Крок 5. Деталізуємо дію заміни елементів.

Псевдокод алгоритму

Початок

Ввід n, m

mas_in(A, n, m)

mas_out(A, n, m)

find_X(A, n, m, index_x, index_y)

X = A[index_x][index_y]

arg = find_arg(A, n, m)

replace(A, n, m, arg, index_x, index_y)

mas_out(A, n, m)

Кінець

Підпрограми:

mas_in(mas[], n, m)

для i < n **повторити**

для j < m **повторити**

mas[i][j] = (double)(rand()) / RAND_MAX * 10

все повторити

mas2[i] = sum

все повторити

mas_out(mas[], n, m)

для i < n **повторити**

для j < m **повторити**

вивід mas[i][j]

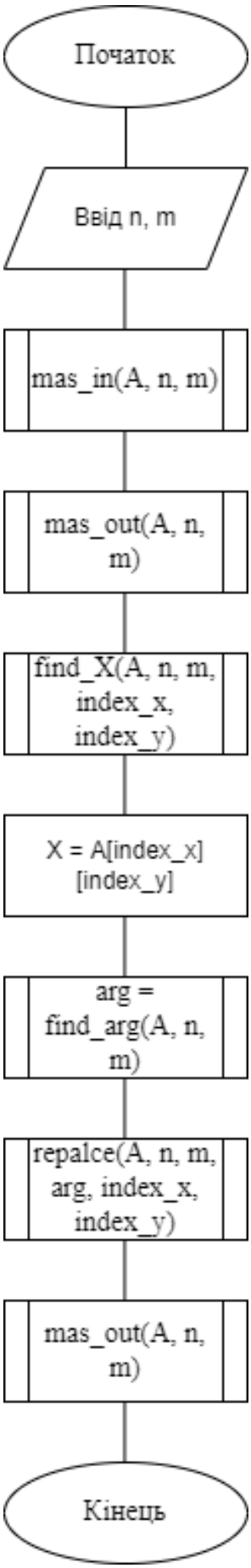
все повторити

все повторити

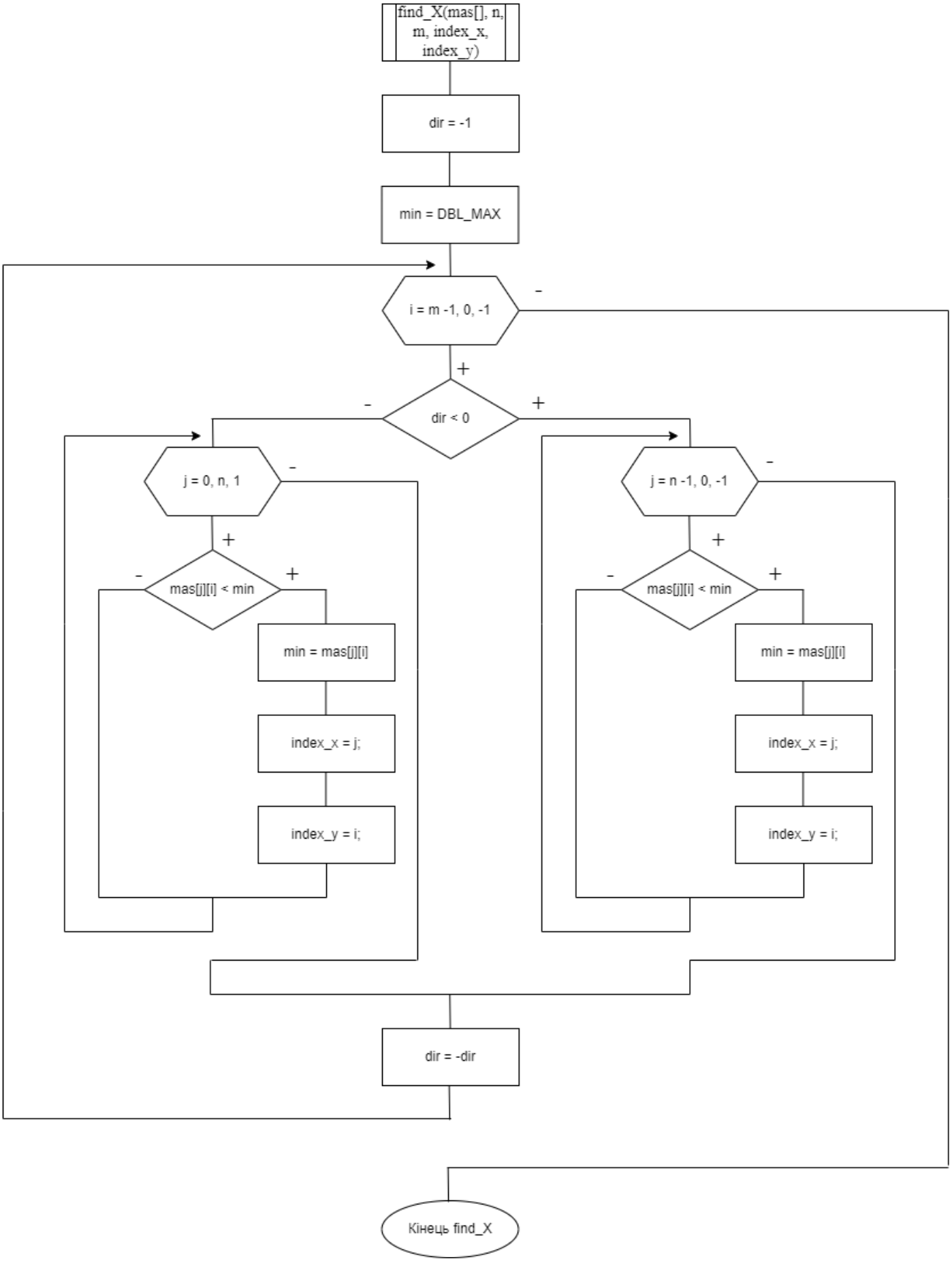
```
find_X(mas[], n, m, index_x, index_y)
    dir = -1
    min = DBL_MAX
    для i від m - 1 до 0 повторити
        якщо dir < 0
            для j від n - 1 до 0 повторити
                якщо mas[j][i] < min
                    min = mas[j][i]
                    index_x = j
                    index_y = i
                все якщо
            все повторити
        інакше
            для j від 0 до n - 1 повторити
                якщо mas[j][i] < min
                    min = mas[j][i]
                    index_x = j
                    index_y = i
                все якщо
            все повторити
        все якщо
        dir = -dir
    все повторити
```

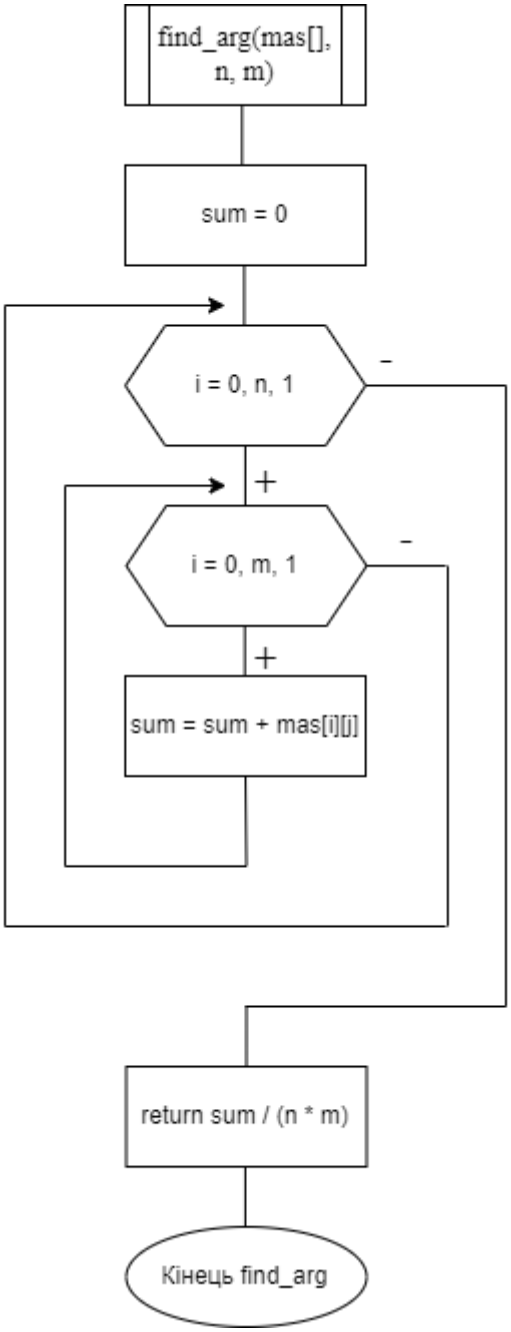
```
find_arg(mas[], n, m)
    sum = 0
    для i < n повторити
        для j < m повторити
            sum = sum + mas[i][j]
        все повторити
    все повторити
    повернути sum / (n * m)

replace(mas[], n, m, arg, index_x, index_y)
    для i < n повторити
        для j від i +1 до m повторити
            якщо mas[i][j] > arg
                znach = mas[index_x][index_y]
                mas[index_x][index_y] = mas[i][j]
                mas[i][j] = znach
                повернути 0
            все якщо
        все повторити
    все повторити
    повернути 0
```

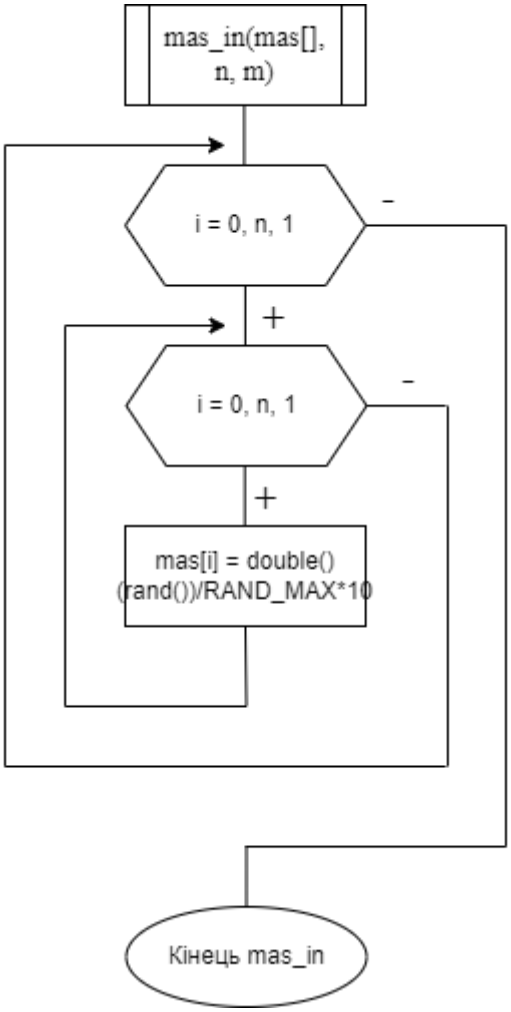
Блок схема алгоритму

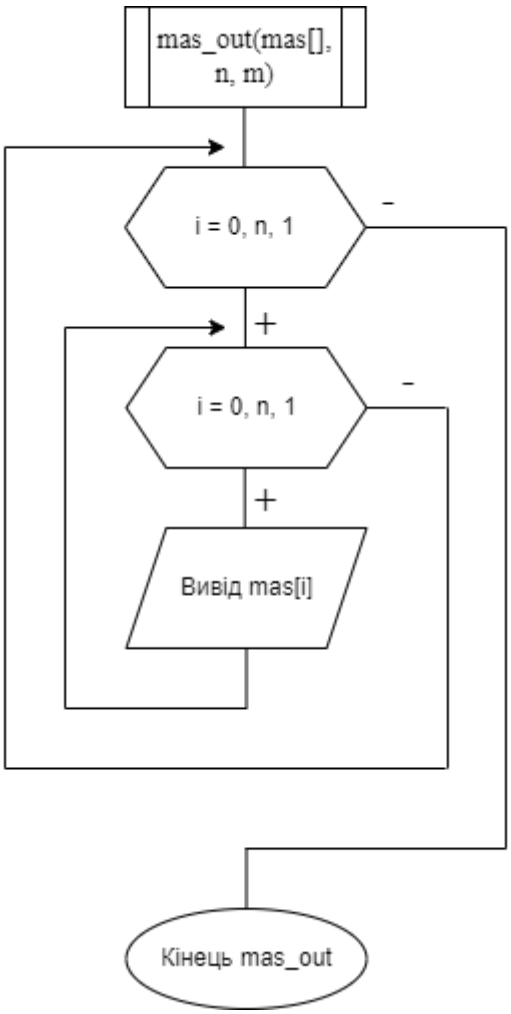


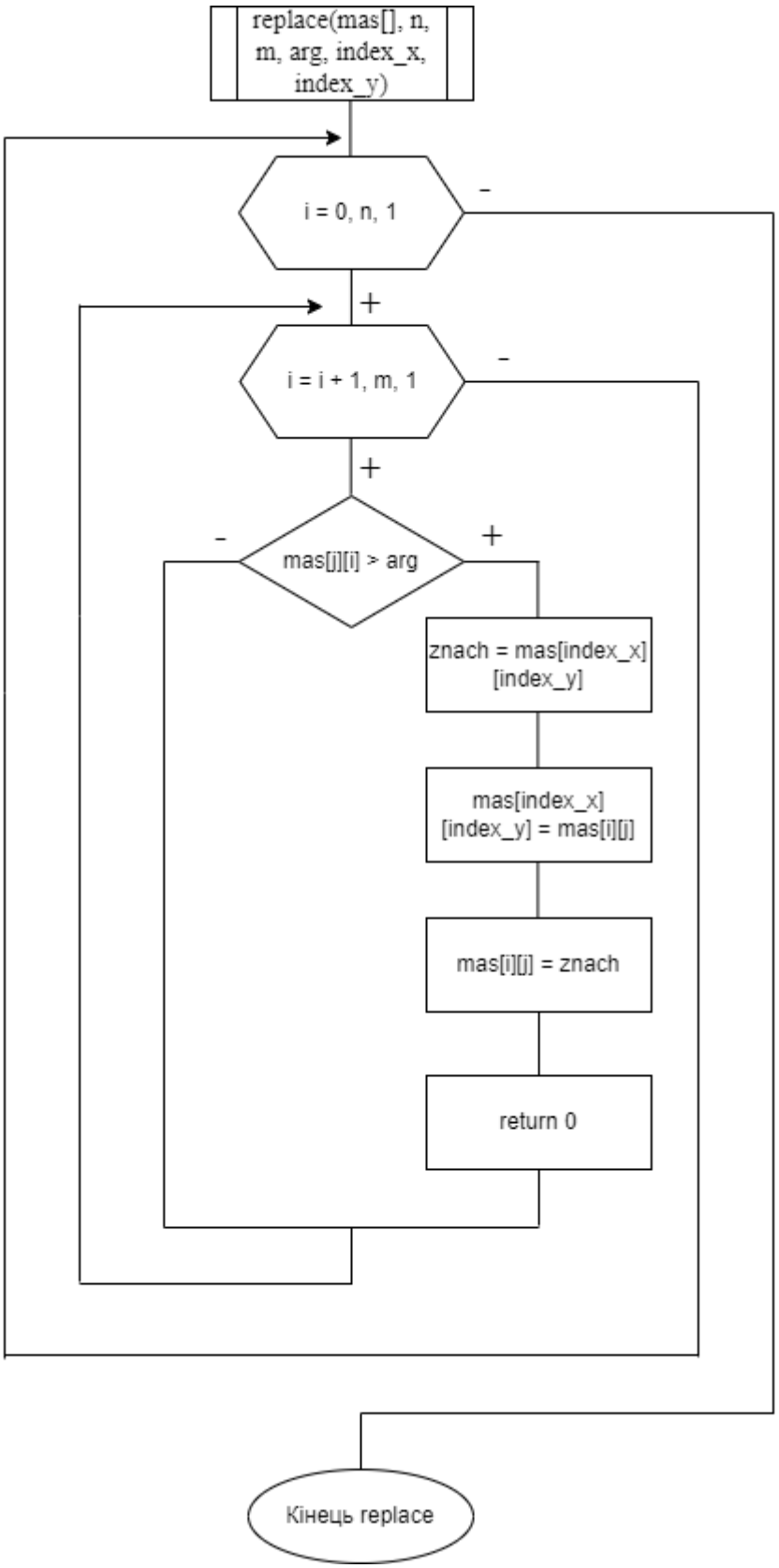
Підпрограми




find_arg(mas[], n, m)
sum = 0
i = 0, n, 1
-
+
i = 0, m, 1
-
+
sum = sum + mas[i][j]
return sum / (n * m)
Кінець find_arg


mas_in(mas[], n, m)
i = 0, n, 1
-
+
i = 0, n, 1
-
+
mas[i] = double() (rand())/RAND_MAX*10
Кінець mas_in


mas_out(mas[], n, m)
i = 0, n, 1
-
+
i = 0, n, 1
-
+
Вивід mas[i]
Кінець mas_out


replace(mas[], n, m, arg, index_x, index_y)
i = 0, n, 1
-
+
i = i + 1, m, 1
-
+
mas[j][i] > arg
-
+
znach = mas[index_x] [index_y]
mas[index_x] [index_y] = mas[i][j]
mas[i][j] = znach
return 0
Кінець replace

Код програми

```
#include <bits/stdc++.h>

using namespace std;


void mas_in( double *mas[], int n, int m){
	srand(time(NULL));
	for(int i = 0; i < n; i++){
		for(int j = 0; j < m; j++){
			mas[i][j] = (double)(rand()) / RAND_MAX * 10;
		}
	}
}


void mas_out( double *mas[], int n, int m){
	for(int i = 0; i < n; i++){
		for(int j = 0; j < m; j++){
			cout << fixed << setprecision(2) << mas[i][j] << " ";
		}
		cout << endl;
	}
	cout << endl;
}
```

```
void find_X( double *mas[], int n, int m, int &index_x, int &index_y){
	int dir = -1;
	double min = DBL_MAX;
	for(int i = m - 1; i >= 0; i--){
		if(dir < 0){
			for(int j = n - 1; j >= 0; j--){
				if(mas[j][i] < min){
					min = mas[j][i];
					index_x = j;
					index_y = i;
				}
			}
		}
		else{
			for(int j = 0; j < n; j++){
				if(mas[j][i] < min){
					min = mas[j][i];
					index_x = j;
					index_y = i;
				}
			}
		}
```

```
            dir = -dir;
        }
}

double find_arg( double *mas[], int n, int m){
        double sum = 0;
        for(int i = 0; i < n; i++){
            for(int j = 0; j < m; j++){
                sum += mas[i][j];
            }
        }
        return sum/(n * m);
}

double replace(double *mas[], int n, int m, double arg, int index_x, int index_y){
        for(int i = 0; i < n; i++){
            for(int j = i + 1; j < m; j++){
                if(mas[i][j] > arg){
                    double znach = mas[index_x][index_y];
                    mas[index_x][index_y] = mas[i][j];
                    mas[i][j] = znach;
                    return 0;
                }
```

```
            }
        }
        return 0;
}


int main(){
        int n, m;
        cout << "Enter size of mas: ";
        cin >> n >> m;
        double A[n][m];
        double *a[n];
        for(int i = 0; i < n; i++){
                a[i] = &A[i][0];
        }
        mas_in(a, n, m);
        cout << "MAS: " << endl;
        mas_out(a, n, m);


        int index_x, index_y;
        find_X(a, n, m, index_x, index_y);
        double X = A[index_x][index_y];
        double arg = find_arg(a, n, m);
        replace(a, n, m, arg, index_x, index_y);
```

```
    cout << "X: " << X << endl;
    cout << "Index: " << index_x << " " << index_y << endl;
    cout << "Arg: " << arg << endl;
    cout << endl << "Replace mas: " << endl;
    mas_out(a, n, m);
}
```

Випробування програми

```
Enter size of mas: 2 2
MAS:
5.30 0.21
6.80 3.01

X: 0.21
Index: 0 1
Arg: 3.83

Replace mas:
5.30 0.21
6.80 3.01
```

```
Enter size of mas: 3 3
MAS:
5.27 8.53 7.80
5.36 4.00 6.74
9.68 1.08 0.12

X: 0.12
Index: 2 2
Arg: 5.40

Replace mas:
5.27 0.12 7.80
5.36 4.00 6.74
9.68 1.08 8.53
```

Висновки

Ми дослідили обходу масивів та набули практичних навичок їх використання під час складання програмних специфікацій.

У результаті виконання лабораторної роботи ми отримали алгоритм для заміни мінімального елементу із елементом над головною діагоналлю, яий більший за середнє арифметичне : визначили основні дії, деталізували дію ініціювання двомірного масиву, дію визначення мінімального елементу, дію визначення середнього арифметичного та дію заміни елементів