

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 No 8 з дисципліни
«Основи програмування 1.
Базові конструкції»
«Багатовимірні масиви»

Варіант 7

Виконав студент ІП-1407 Грицина Діана Русланівна (шифр, прізвище,
ім'я, по батькові)

Перевірів _____ (прізвище, ім'я, по
батькові)

Київ 2021

Лабораторна робота No 8

Тема: Багатовимірні масиви.

Мета: Опанувати технологію використання двовимірних масивів даних (матриць), навчитися розробляти алгоритми та програми із застосуванням матриць.

Хід роботи

Задача

7. Задані три цілочисельні матриці розмірності $m \times m$, що містять від'ємні елементи. З'ясувати, яка з цих матриць має найменшу суму модулів діагональних елементів тих рядків, що починаються з від'ємного елемента. Упорядкувати рядки визначеної матриці за збільшенням значень її елементів.

Постановка задачі: Для того щоб розмір матриці можна було задавати, у програмі потрібно використовувати двовимірні динамічні масиви. До кожного елемента матриці будемо звертатися по індексу $a[i][j]$. Оскільки завдання потребує використання 3 матриць, доречно буде винести їх ініціалізацію та обробку в окремі функції. Далі потрібно порівняти суми елементів, що відбираються за умовою задачі. Для матриці, що має найменшу таку суму, застосуємо перестановку елементів кожного рядка.

Математична модель:

Назва	Тип	Призначення
Функція <code>make_matrix()</code>	Void	Генерація елементів матриці випадковим чином
Функція <code>output_matrix()</code>	Void	Виведення матриці
Функція <code>sum_abs_find()</code>	Int	Знаходження суми потрібних елементів
Функція <code>change_el()</code>	Void	Упорядкування рядків

Код програми на C++

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <cmath>
using namespace std;
void make_matrix(int **a, int m);
void output_matrix(int **a, int m);
int sum_abs_find(int **a, int m);
void change_el(int **a, int m);
int main(){
    srand((unsigned int)time(NULL));
    int m;
    cout<<"Enter m ";
    cin>>m; //Введення розмірності матриці
    int **a, **b, **c; //Створення покажчика на масив покажчиків
    a = new int*[m]; //Створення масиву покажчиків
    b = new int*[m];
    c = new int*[m];
    for(int i=0; i<m; i++){
        a[i] = new int [m]; //Створення рядків матриці
        b[i] = new int [m];
        c[i] = new int [m];
    }
    make_matrix(a, m); //Ініціалізація матриці
    make_matrix(b, m);
    make_matrix(c, m);
    output_matrix(a, m); //Виведення матриці
    output_matrix(b, m);
    output_matrix(c, m);
    int sum_a = sum_abs_find(a, m); //Пошук суми елементів у матриці за умовою задачі
    int sum_b = sum_abs_find(b, m);
    int sum_c = sum_abs_find(c, m);
    cout<<"Sum a "<<sum_a<<"\n";
    cout<<"Sum b "<<sum_b<<"\n";
    cout<<"Sum c "<<sum_c<<"\n";
    int smaller = (sum_a < sum_b) ? (sum_c<sum_a ? sum_c : sum_a) : (sum_c<sum_b ? sum_c :
        sum_b);
    cout<<"The smallest value: "<<smaller<<"\n";
```

```

    if(smaller == sum_a){
        change_el(a, m);
        output_matrix(a, m);
    }else if(smaller == sum_b){
        change_el(b, m);
        output_matrix(b, m);
    }else{
        change_el(c, m);
        output_matrix(c, m);
    }
    for(int i=0; i<m; i++){
        delete []a[i]; //Звільнення пам'яті від кожного рядка матриці
        delete []b[i];
        delete []c[i];
    }
    delete []a; //Звільнення пам'яті від масиву покажчиків
    delete []b;
    delete []c;
}

void make_matrix(int **a, int m){
    for(int i=0; i<m; i++){
        for(int j=0; j<m; j++){
            a[i][j] = rand()%41-20;
        }
    }
}

void output_matrix(int **a, int m){
    for(int i=0; i<m; i++){
        for(int j=0; j<m; j++){
            cout<<a[i][j]<<"\t";
        }
        cout<<"\n";
    }
    cout<<"\n";
}

int sum_abs_find(int **a, int m){
    int sum = 0;
    for(int i=0; i<m; i++){
        if(a[i][0]<0){
            int j = i;
            sum += abs(a[i][j]);
        }
    }
    return sum;
}

void change_el(int **a, int m){
    for(int i=0; i<m; i++){
        for(int j=0; j<m-1; j++){
            for(int c=0; c<m-1; c++){
                if(a[i][c]>a[i][c+1]){
                    int var = a[i][c];
                    a[i][c] = a[i][c+1];
                    a[i][c+1] = var;
                    //std::swap(a[i][c], a[i][c+1]);
                }
            }
        }
    }
}

```

Результат

```
Enter m 3
2  -9  15
14 13  0
10 17 -15

-5 -9 12
14 6  9
13 0 -12

-10 19 -7
-4 -4 -9
-5 5 -2

Sum a 0
Sum b 5
Sum c 16
The smallest value: 0
-9 2 15
0 13 14
-15 10 17

Program ended with exit code: 0
```

```
Enter m 7
18 1 6 18 -10 16 -13
6 6 9 18 -8 -13 -6
-12 20 -11 20 7 6 -7
16 19 -13 -4 -6 15 -15
-15 0 -8 14 -20 -5 4
20 -9 2 -13 -15 -7 -5
0 1 13 -3 12 12 16

7 -4 -2 17 3 9 -18
-12 9 -13 -17 4 -8 -11
10 -6 -8 -6 16 -20 -14
-9 3 18 6 -5 -1 -9
9 14 -10 -5 3 -15 -6
-12 -15 2 8 17 -20 -14
-5 4 -6 -11 19 -2 -6

-5 -2 5 11 -2 -5 3
0 3 12 16 8 16 16
-8 10 19 4 16 16 6
-5 -5 3 -12 -3 1 5
6 6 -20 -7 -12 17 3
-4 19 -10 -14 6 -19 -8
-19 -17 1 6 -8 -7 11

Sum a 31
Sum b 41
Sum c 66
The smallest value: 31
-13 -10 1 6 16 18 18
-13 -8 -6 6 6 9 18
-12 -11 -7 6 7 20 20
-15 -13 -6 -4 15 16 19
-20 -15 -8 -5 0 4 14
-15 -13 -9 -7 -5 2 20
-3 0 1 12 12 13 16

Program ended with exit code: 0
```

Висновок: У роботі досліджено використання двовимірних динамічних масивів при роботі з матрицями. Передача таких масивів у функцію виконується тільки за адресою, де виконується ініціалізація матриці, виведення та обробка її елементів.