МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1**

**з дисципліни «Основи програмної інженерії»**

**на тему: «Модульне програмування. Розробка програми методом низхідного програмування.»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Куроп'ятник О. С.

Дніпро, 2019

**Лабораторна робота №1**

**Тема.** Модульне програмування. Розробка програми методом низхідного програмування

**Мета.** Отримати практичні навички розробки програм за методом низхідного програмування.

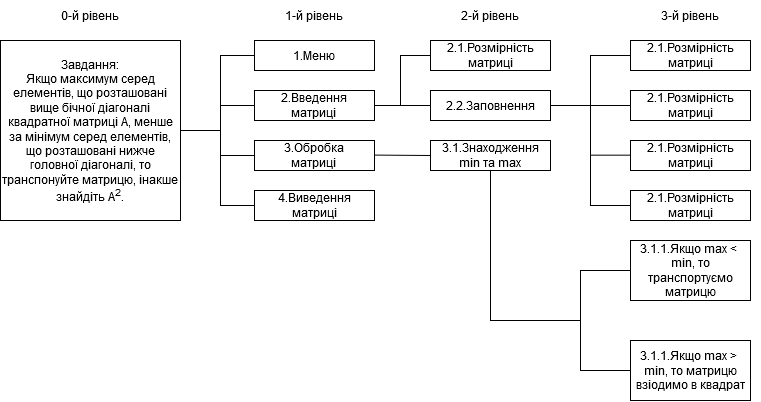
**Постановка задачі**

1.Загальна постановка: розробити програму для обробки динамічної матриці за індивідуальним завданням. Програма повинна мати текстове меню. Данні для виконання програми вводяться з клавіатури або генеруються випадкові значення.

2.Індивідуальна постановка:

|  |  |
| --- | --- |
| Варіант | Завдання |
| 18. | Якщо максимум серед елементів, що розташовані вище бічної діагоналі квадратної матриці А, менше за мінімум серед елементів, що розташовані нижче головної діагоналі, то транспонуйте матрицю, інакше знайдіть А2. |

**Схема модульної структури програми**

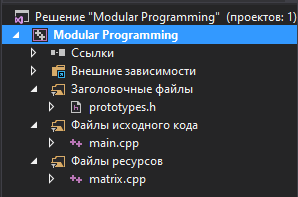


**Розробка програми за методом низхідного програмування**

Реалізуємо 0-й рівень модульної структури програми: створимо головний модуль – функцію main() і модулі-заглушки першого рівня структури програми (функції New2DArray(), Delete2DArray(), PrintArray(), FillingArray() для модулів 1-4 відповідно).

Після реалізації основних заготовок (0 та 1 рівня) розробляемо модулі знаходження мінімального та максимального числа (FindingMin(), FindingMax()), і потім вже порівнюємо максимальне та мінімальне число та в залежності від результату ми транспортуємо матрицю (NewTransposeArray()) або зводимо її в квадрат (SquareArray()).

Розробимо файлову структуру проекту. Головний модуль розмістимо у файлі Pr1\_1.cpp. Модулі, пов’язані з консольним введенням-виведенням і обробкою матриці (модулі 1, 2, 3, 4) розмістимо у двох загальних файлах – заготовочному файлі func.h і файлі реалізації func.cpp. А також в файлі func.h запишемо структуру матриці.



Файлова структура проекту

Файл main.cpp

#include <iostream>

#include "prototypes.h"

#include "Windows.h"

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

std::system("menu.txt");

int number;

do

{

std::system("cls");

std::cout << "1. Заповнювати матрицю з клавіатури" << std::endl;

std::cout << "2. Випадкове заповнення матриці" << std::endl;

std::cout << "3. Вихід" << std::endl;

while (!(std::cin >> number) || (std::cin.peek() != '\n')) //перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

system("cls");

std::cerr << "Помилка вводу! Повторіть ввод. Номер меню повинен бути цілим числом." << std::endl;

std::cout << std::endl;

std::cout << "1. Заповнювати матрицю з клавіатури" << std::endl;

std::cout << "2. Випадкове заповнення матриці" << std::endl;

std::cout << "3. Вихід" << std::endl;

}

switch (number)

{

case 1:

{

std::system("cls");

int size;

std::cout << "Введіть розмір матриці: ";

while (!(std::cin >> size) || (std::cin.peek() != '\n') || !(size > 1)) //перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

std::cerr << "Помилка вводу! Повторіть ввод. Розмір матриці повинен бути додатним числом и більше 1." << std::endl;

std::cout << std::endl;

std::cout << "Введіть розмір матриці: ";

}

int n = 1;

int \*\*Array = New2DArray(size);

for (int i = 0; i < size; i++)//заповнення масива нулями

{

for (int j = 0; j < size; j++)

{

Array[i][j] = 0;

}

}

for (int m = 0; m < size; m++)//заповнення масива з клавіатури

{

for (int n = 0; n < size; n++)

{

std::system("cls");

PrintArray(Array, size);

std::cout << "arr[" << m << "][" << n << "] = ";

std::cin >> Array[m][n];

}

}

std::system("cls");

PrintArray(Array, size);

int max;

max = FindingMax(Array, size);

//std::cout << "Максимальне число " << max << std::endl;

int min;

min = FindingMin(Array, size);

//std::cout << "Мінімальне число " << min << std::endl;

if (max < min)

{

std::cout << "Так як максимальне число меньше мінімального, то ми транспонуємо матрицю" << std::endl;

int\*\* tmparr = NewTransposeArray(Array, size);

PrintArray(tmparr, size);

Delete2DArray(tmparr, size);

}

else

{

std::cout << "Так як максимальне число більше мінімального, то ми матрицю підносимо у квадрат" << std::endl;

std::cout << "Матриця в квадраті:" << std::endl;

SquareArray(Array, size);

PrintArray(Array, size);

}

Delete2DArray(Array, size);

std::system("pause");

}

break;

case 2:

{

std::system("cls");

int size;

std::cout << "Введіть розмір матриці: ";

while (!(std::cin >> size) || (std::cin.peek() != '\n') || !(size > 1)) //перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

std::cerr << "Помилка вводу! Повторіть ввод. Розмір матриці повинен бути цілим числом та більшим за 1." << std::endl;

std::cout << std::endl;

std::cout << "Введіть розмір матриці: ";

}

int \*\*Array = New2DArray(size);

FillingArray(Array, size);

PrintArray(Array, size);

int max;

max = FindingMax(Array, size);

//std::cout << "Максимальне число " << max << std::endl;

int min;

min = FindingMin(Array, size);

//std::cout << "Мінімальне число " << min << std::endl;

if (max < min)

{

std::cout << "Так як максимальне число меньше мінімального, то ми транспонуємо матрицю" << std::endl;

std::cout << "Транспортована матриця:" << std::endl;

int\*\* tmparr = NewTransposeArray(Array, size);

PrintArray(tmparr, size);

Delete2DArray(tmparr, size);

}

else

{

std::cout << "Так як максимальне число більше мінімального, то ми матрицю підносимо у квадрат" << std::endl;

std::cout << "Матриця в квадраті:" << std::endl;

SquareArray(Array, size);

PrintArray(Array, size);

}

Delete2DArray(Array, size);

std::system("pause");

}

break;

case 3:

return 0;

default:

std::system("cls");

std::cout << "Введено невірний номер меню!" << std::endl << std::endl;

std::system("pause");

break;

}

} while (number != 3);

}

Файл prototypes.h

int\*\* New2DArray(const int size);

void Delete2DArray(int\*\* Arr, const int size);

void PrintArray(int \*\*arr, const int size);

void FillingArray(int \*\*arr, const int size);

int FindingMax(int \*\*arr, const int size);

int FindingMin(int \*\*arr, const int size);

int\*\* NewTransposeArray(int \*\*arr, const int size);

void SquareArray(int \*\*arr, const int size);

Файл matrix.h

#include <iostream>

#include <random>

#include "prototypes.h"

int\*\* New2DArray(const int size)//створення масива

{

int\*\* Arr = new int\*[size];

for (int i = 0; i < size; ++i)

Arr[i] = new int[size];

return Arr;

}

void Delete2DArray(int\*\* Arr, const int size)//виділення масива

{

for (int i = 0; i < size; ++i)

delete[] Arr[i];

delete[] Arr;

}

void PrintArray(int\*\* arr, const int size)//друк масива

{

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

std::cout << arr[i][j] << "\t";

}

std::cout << std::endl;

}

}

void FillingArray(int \*\*arr, const int size)//заповнення масива випадковими числами

{

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<int> dis(0, 10);

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

arr[i][j] = dis(gen);

}

}

}

int FindingMax(int \*\*arr, const int size)//знаходження максимального числа

{

int max, m = 1;

max = arr[0][0];

for (int i = 0; i < size - 1; i++, m++)

{

for (int j = 0; j < size - m; j++)

{

if (arr[i][j] > max)

{

max = arr[i][j];

}

}

}

return max;

}

int FindingMin(int \*\*arr, const int size)//знаходження мінімального числа

{

int min, m = 1;

min = arr[1][0];

for (int i = 1; i < size - 1; i++, m++)

{

for (int j = 0; j < m; j++)

{

if (arr[i][j] < min)

{

min = arr[i][j];

}

}

}

return min;

}

int\*\* NewTransposeArray(int \*\*arr, const int size)//транспортування матриці

{

int\*\* res = New2DArray(size);

for (int i = 0; i < size; ++i)

{

for (int j = 0; j < size; ++j)

{

res[j][i] = arr[i][j];

}

}

return res;

}

void SquareArray(int \*\*arr, const int size)//матриця в квадраті

{

int \*\*temp = New2DArray(size);

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

temp[i][j] = 0;

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

for (int k = 0; k < size; k++)

{

temp[i][j] += arr[i][k] \* arr[k][j];

}

}

}

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

arr[i][j] = temp[i][j];

temp[i][j] = 0;

}

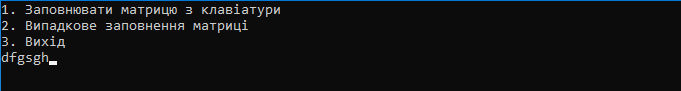
}

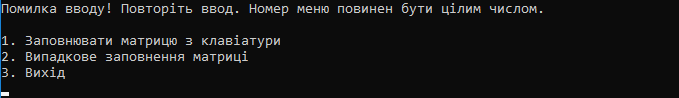
}

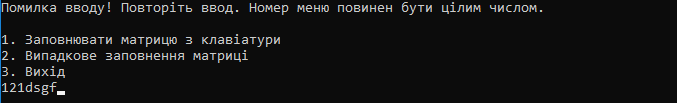
**Результати виповнення програми**

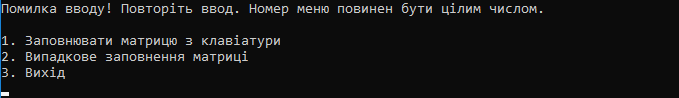
1. Введення номера меню

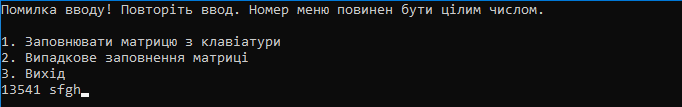
а) Некоректні

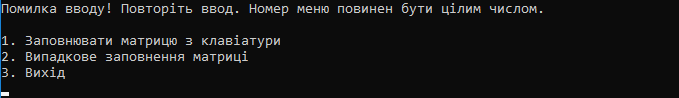


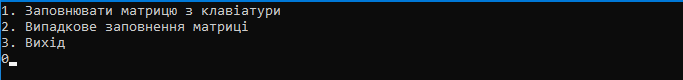


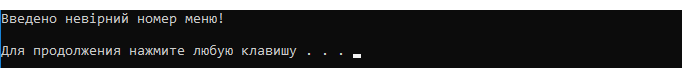




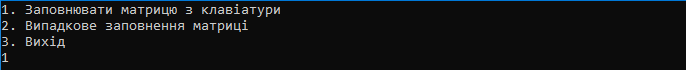


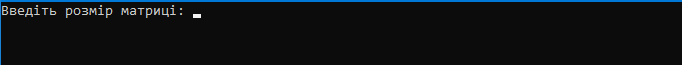






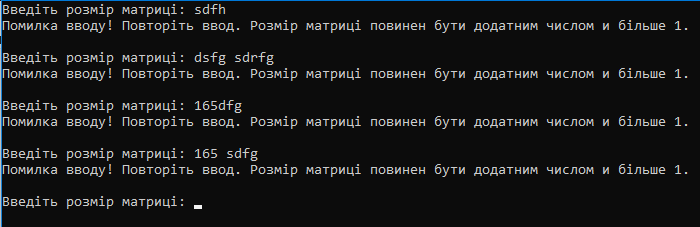
б) Коректні



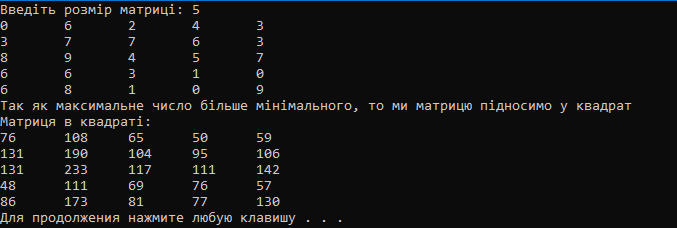


2)Розмір матриці

а) Некоректні

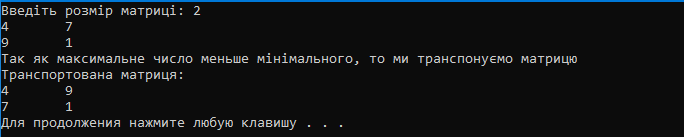


б)Коректні

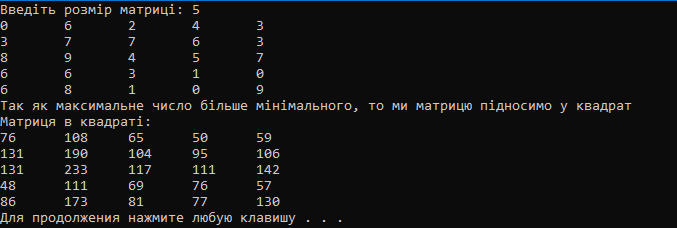


3)Результати порівнянь

а)max < min



б)max > min



Висновок: після перевірки набором тестів програма не відобразила ніяких недоліків, умова індивідуального завдання було виконано. Для розробки програми використовувалися метод низхідного програмування і покрокової деталізації. Використання методів для розробки значно полегшило розробку алгоритму, адже коли розкладаєш велику задачу на багато малих задач велика ймовірність досягти успіху. Менше помилок і більше концентрації на окремих маленьких задачах полегшує вирішення поставленого великого питання. Отже, поставлена задача була виконана.