**ЗВІТ**

**до лабораторної роботи № < 7.3 >**

**« Опрацювання динамічних багатовимірних масивів »**

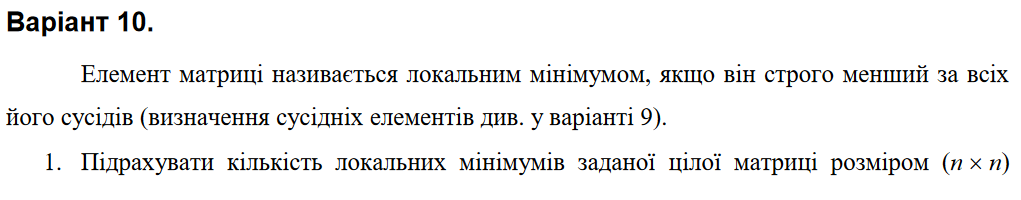
**з дисципліни**

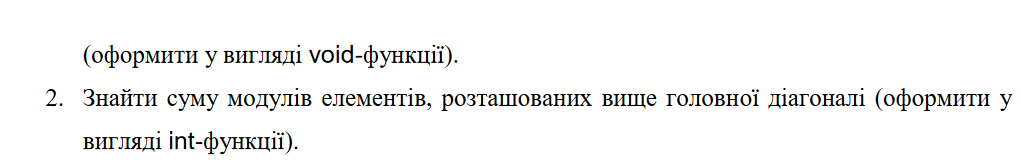
**«Алгоритмізація та програмування»**

**студента групи ІН-105Б**

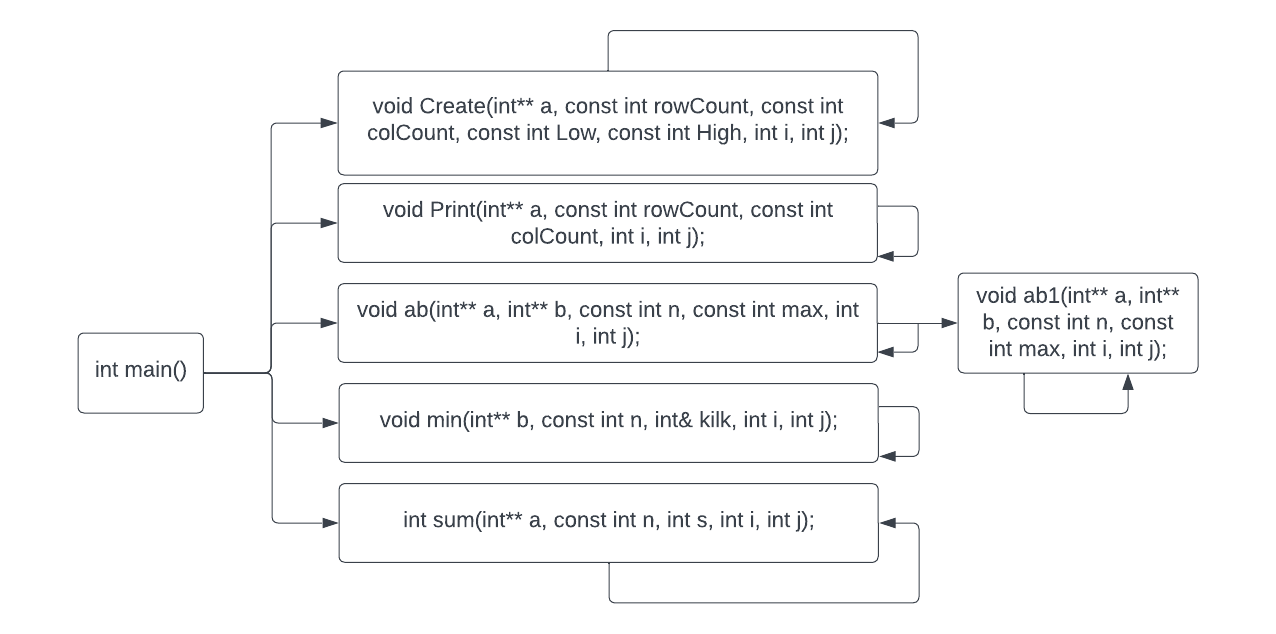
**Горанова Анастаса**

**Умова завдання:**





**Структурна схема:**



**Текст програми:**

// 7.3REC.cpp

// < Горанов Анастас >

// Лабораторна робота № 7.3REC

// Опрацювання динамічних багатовимірних масивів

// Варіант 10

#include <iostream>

#include <iomanip>

#include <time.h>

using namespace std;

void Create(int\*\* a, const int rowCount, const int colCount, const int Low, const int High, int i, int j);

void Print(int\*\* a, const int rowCount, const int colCount, int i, int j);

void ab1(int\*\* a, int\*\* b, const int n, const int max, int i, int j);

void ab(int\*\* a, int\*\* b, const int n, const int max, int i, int j);

void min(int\*\* b, const int n, int& kilk, int i, int j);

int sum(int\*\* a, const int n, int s, int i, int j);

int main()

{

srand((unsigned)time(NULL));

int Low = -10;

int High = 10;

int n, kilk;

cout << "rowCount = ";

cin >> n;

int\*\* a = new int\* [n];

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

a[i] = new int[n];

int\*\* b = new int\* [n + 2];

for (int i = 0; i < n + 2; i++)

b[i] = new int[n + 2];

Create(a, n, n, Low, High, 0, 0);

Print(a, n, n, 0, 0);

ab(a, b, n + 2, High + 1, 0, 0);

Print(b, n + 2, n + 2, 0, 0);

kilk = 0;

min(b, n + 2, kilk, 1, 1);

cout << "kilkist = " << kilk << endl;

cout << "sum = " << sum(a, n, 0, 0, 1) << endl;

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

{

delete[] a[i];

delete[] b[i];

}

delete[] a;

delete[] b;

return 0;

}

void Create(int\*\* a, const int rowCount, const int colCount, const int Low, const int High, int i, int j)

{

a[i][j] = Low + rand() % (High - Low + 1);

if (j < colCount - 1)

Create(a, rowCount, colCount, Low, High, i, j + 1);

else

if (i < rowCount - 1)

Create(a, rowCount, colCount, Low, High, i + 1, 0);

}

void Print(int\*\* a, const int rowCount, const int colCount, int i, int j)

{

cout << setw(4) << a[i][j];

if (j < colCount - 1)

Print(a, rowCount, colCount, i, j + 1);

else

if (i < rowCount - 1)

{

cout << endl;

Print(a, rowCount, colCount, i + 1, 0);

}

else

cout << endl << endl;

}

void ab1(int\*\* a, int\*\* b, const int n, const int max, int i, int j)

{

b[i][j] = a[i - 1][j - 1];

if (j < n - 2)

ab1(a, b, n, max, i, j + 1);

else

if (i < n - 2)

ab1(a, b, n, max, i + 1, 1);

}

void ab(int\*\* a, int\*\* b, const int n, const int max, int i, int j)

{

b[i][j] = max;

if (j < n - 1)

ab(a, b, n, max, i, j + 1);

if (i < n - 1)

ab(a, b, n, max, i + 1, 0);

else

ab1(a, b, n, max, 1, 1);

}

void min(int\*\* b, const int n, int& kilk, int i, int j)

{

if (

(b[i][j] < b[i][j - 1]) &&

(b[i][j] < b[i - 1][j]) &&

(b[i][j] < b[i][j + 1]) &&

(b[i][j] < b[i + 1][j])

)

{

kilk++;

}

if (j < n - 1)

min(b, n, kilk, i, j + 1);

else

if (i < n - 1)

min(b, n, kilk, i + 1, 1);

}

int sum(int\*\* a, const int n, int s, int i, int j)

{

if (i < n)

{

if (j < n)

{

s += abs(a[i][j]);

sum(a, n, s, i, j + 1);

}

else

sum(a, n, s, i + 1, i + 2);

}

else

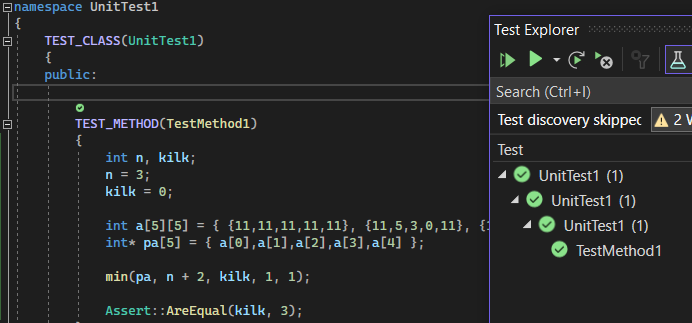
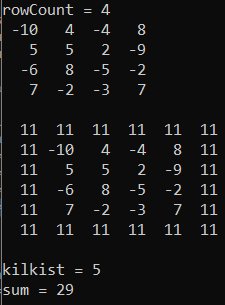
return s;

}

**Посилання на git-репозиторій з проектом:**

*https://github.com/StassNG/7.3REC*

**Результати програми та unit-тесту:**

****

**Висновок:** я навчився опрацьовувати масиви використовуючи рекурсивні функції.