Міністерство освіти та науки України

Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

**ЗВІТ**

З лабораторної роботи №8

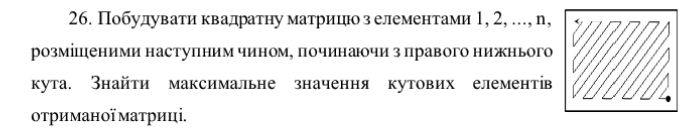
На тему:

«Багатовимірні масиви»

**Виконав:** *Гр. ІС-02, Плостак Ілля Михайлович*

**Прийняв:** *Ст. викладач Проскура Світлана Леонідівна*

1. Умова



2. Псевдокод

**Функция** КвадратЛи(**Целое число** нум)

**Если** ВозвестиВСтепень(ВозвестиВСтепень(дробное(нум), 0.5), 2.0) == ВозвестиВСтепень(дробное(целое(ВозвестиВСтепень(дробное(нум), 0.5))), 2.0):

**Вернуть** Правда

**Иначе**:

**Вернуть** Ложь

**Конец** **условия**

**Конец** **функции**

**Функция** ЗаполнитьДиагональ(**Массив указателей** Матрица, **Целое число** РазмерМатрицы, **Целое число** НомерДиагонали, **Целое число** ПервоеЧисло, **Булевая переменная** Вверх)

**Целое число** нум = ПервоеЧисло

**Если** Вверх:

**Цикл от** i = РазмерМатрицы-1 **до** i >= 0 **c шагом** -1 **Выполнить**:

**Цикл от** j = РазмерМатрицы-1 **до** j >= 0 **c шагом** -1 **Выполнить**:

**Если** i + j + 1 == НомерДиагонали:

Матрица[i][j] = нум--

**Конец условия**

**Конец цикла**

**Конец цикла**

**Иначе**:

**Цикл от** i = 0 **до** i < РазмерМатрицы **c шагом** 1 **Выполнить**:

**Цикл от** j = 0 **до** j < РазмерМатрицы **c шагом** 1 **Выполнить**:

**Если** i + j + 1 == НомерДиагонали:

Матрица[i][j] = нум--

**Конец условия**

**Конец цикла**

**Конец цикла**

**Конец условия**

**Вернуть** нум

**Конец функции**

**Функция** ВыводМатрицы (**Массив указателей** Матрица, **Целое число** РазмерМатрицы)

**Вывод**: Матрица:

**Вывод**: ПЕРЕВОД\_СТРОКИ

**Цикл** **от** i = 0 **до** i < РазмерМатрицы **c шагом** 1 **Выполнить**:

**Цикл от** j = 0 **до** j < РазмерМатрицы **c шагом** 1 **Выполнить**:

**Вывод**: Матрица[i][j]

**Вывод**: ПРОБЕЛ

**Конец цикла**

**Вывод**: ПЕРЕВОД\_СТРОКИ

**Конец цикла**

**Конец Функции**

**Начало**

**Целое число** нум

**Целое число** РазмерМатрицы

**Целый массив указателей** Матрица

**Вывод**: Введите количество чисел для заполнения квадратной матрицы

нум = ВводЦелогоЧисла

РазмерМатрицы = КвадратныйКорень(нум)

Матрица = новый МассивУказателей[РазмерМатрицы]

**Цикл от** i = 0 **до** i < РазмерМассива **c шагом** 1 **Выполнить**:

Матрица[i] = новый МассивЦелых[РазмерМатрицы]

**Конец цикла**

**Цикл от** i = 1 **до** i < 2 \* РазмерМассива **c шагом** 1 **Выполнить**:

**Булевая переменная** Вверх

**Если** i % 2 == 0:

Вверх = Ложь

**Иначе**:

Вверх = Правда

**Конец условия**

нум = ЗаполнитьДиагональ(Матрица, РазмерМатрицы, i, нум, Вверх)

**Конец цикла**

ВыводМатрицы(Матрица, РазмерМатрицы)

**Конец**

3. Виконання на мовi С++

[main.cpp](https://github.com/feedblackg44/kpilabs/blob/master/Lab8/Lab8/main.cpp):

#include <iostream>

#include <math.h>

#include <limits>

#include <locale.h>

#include <iomanip>

#include "feed.h"

#include "functions.h"

using namespace std;

int main()

{

SetDefaults();

LabHeader(8);

int n, // Количество чисел для заполнения матрицы

size; // Размер матрицы

int\*\* Matrix; // Указатель на матрицу

GetMatrixSize(&n, &size);

Matrix = MatrixInit(size);

BuildMatrix(Matrix, n, size);

OutputMatrix("Матрица", Matrix, size);

SystemPause();

return 0;

}

[functions.h](https://github.com/feedblackg44/kpilabs/blob/master/Lab8/Lab8/functions.h):

#pragma once

bool IsSqare(int num);

int FillDiagonal(int\*\* matrix, int size, int diagNum, int firstNum, bool up);

void OutputMatrix(const char matrixName[], int\*\* matrix, int size);

void GetMatrixSize(int\* num, int\* size);

int\*\* MatrixInit(int size);

void BuildMatrix(int\*\* Matrix, int n, int size);

[functions.cpp](https://github.com/feedblackg44/kpilabs/blob/master/Lab8/Lab8/functions.cpp):

#include <iostream>

#include <math.h>

#include <limits>

#include <locale.h>

#include <iomanip>

#include "feed.h"

#include "functions.h"

using namespace std;

bool IsSqare(int num)

{

if (pow(pow(double(num), 0.5), 2.0) == pow(double(int(pow(double(num), 0.5))), 2.0))

{

return true;

}

else

{

return false;

}

}

int FillDiagonal(int\*\* matrix, int size, int diagNum, int firstNum, bool up)

{

int num = firstNum;

if (up)

{

for (int i = size-1; i >= 0; i--)

{

for (int j = size-1; j >= 0; j--)

{

if (i + j + 1 == diagNum)

{

matrix[i][j] = num--;

}

}

}

}

else

{

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

{

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

{

if (i + j + 1 == diagNum)

{

matrix[i][j] = num--;

}

}

}

}

return num;

}

void OutputMatrix(const char matrixName[], int\*\* matrix, int size)

{

cout << matrixName << ":" << endl;

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

{

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

{

if (matrix[i][j] < 10)

cout << " ";

else if (matrix[i][j] >= 10 && matrix[i][j] < 100)

cout << " ";

cout << matrix[i][j] << " ";

}

cout << endl;

}

cout << endl;

}

void GetMatrixSize(int \*num, int \*size)

{

do

{

\*num = GetInt("Введите количество чисел для заполнения квадратной матрицы:", "Число введено неправильно!");

if (!IsSqare(\*num))

{

cout << "Так как матрица квадратная, число должно быть полным квадратом!" << endl;

}

} while (!IsSqare(\*num));

(\*size) = pow(double(\*num), 0.5);

}

int\*\* MatrixInit(int size)

{

int\*\* Matrix;

Matrix = new int\* [size];

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

{

Matrix[i] = new int[size];

}

return Matrix;

}

void BuildMatrix(int\*\* Matrix, int num, int size)

{

for (int i = 1; i < 2 \* size; i++)

{

bool up = i % 2 ? true : false;

num = FillDiagonal(Matrix, size, i, num, up);

}

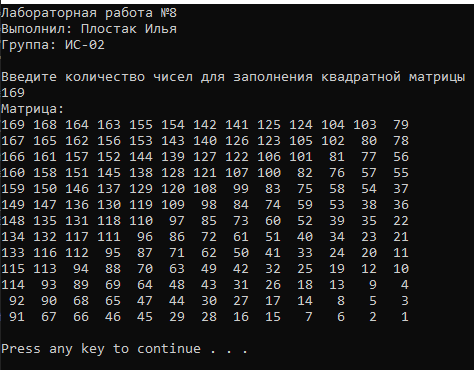
}

[feed.h](https://github.com/feedblackg44/kpilabs/blob/master/include/feed.h)

[feed.cpp](https://github.com/feedblackg44/kpilabs/blob/master/include/feed.cpp)

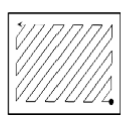
4. Результат виконання програми

(Visual Studio 2019 Community Edition)



5. Перевiрка результатiв не потрiбна.

6. Висновок

В цій лабораторній роботі ми опанували технологію використання двовимірних масивів даних (матриць), навчилися розробляти алгоритми та програми із застосуванням матриць. За допомогою двовимірного масиву та вкладеного циклу ми заповнили матрицю елементами 1, 2, 3, …, n, розміщеними таким чином: , де n – число, яке вводить користувач, та вивели цю матрицю в консоль за допомогою вкладеного циклу. Ми використали окрему підпрограму IsSqare(), яка приймає на вхід ціле число та повертає true чи false залежно від того чи є вхідне число повним квадратом чи ні, ще одну окрему підпрограму FillDiagonal(), яка приймає на вхід подвійний вказівник (матриця), три цілочисельних значення (розмір квадратної матриці, номер діагоналі та значення з якого починати заповнювання) та булєву змінну (заповнювати діагональ починаючи знизу чи зверху), та яка повертає наступне значення після останнього на даній діагоналі, для заповнювання кожної діагоналі матриці, а також підпрограму OutputMatrix(), яка приймає на вхід вказівник на першу літеру рядка (ім’я матриці), подвійний вказівник (матрицю) та її розмір, та яка нічого не повертає, для виводу отриманої матриці.

Робота виконана у середовищі: Visual Studio 2019 Community Edition.