

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

Дисциплина: «Программирование»

Отчёт по лабораторной работе № 1

**Лабораторная работа №1. Использование массивов**

**Выполнил студент группы №485:**

**Проверили:**

Иван Григорьевич Корниенко

Алексей Константинович Федин

Санкт-Петербург

2019

# Постановка задачи

Матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива по строкам. Восстановить исходную квадратную матрицу и напечатать по строкам.

# Исходные данные

В качестве исходных данных программа использует вводимое пользователем размер матрицы. В случае выбора пользователем заполнения массива из файла, программа запросит ввести путь к этому файлу, где первым значением должен быть размер матрицы, остальные же числа заполняют матрицу.

# Особые ситуации

* Если размер матрицы, указанный пользователем, меньше двух, то программа запросит ввести его заново.
* Если пользователь при указании пути к файлу будет использовать запрещённые имена, например: con, aux и т.д., то программа попросит ввести путь к файлу заново.
* Если в файле, из которого должен быть заполнен массив, вместо числа будет найден символ, то программа попросит пользователя исправить файл и ввести путь заново.

# Математические методы и алгоритмы решения задач

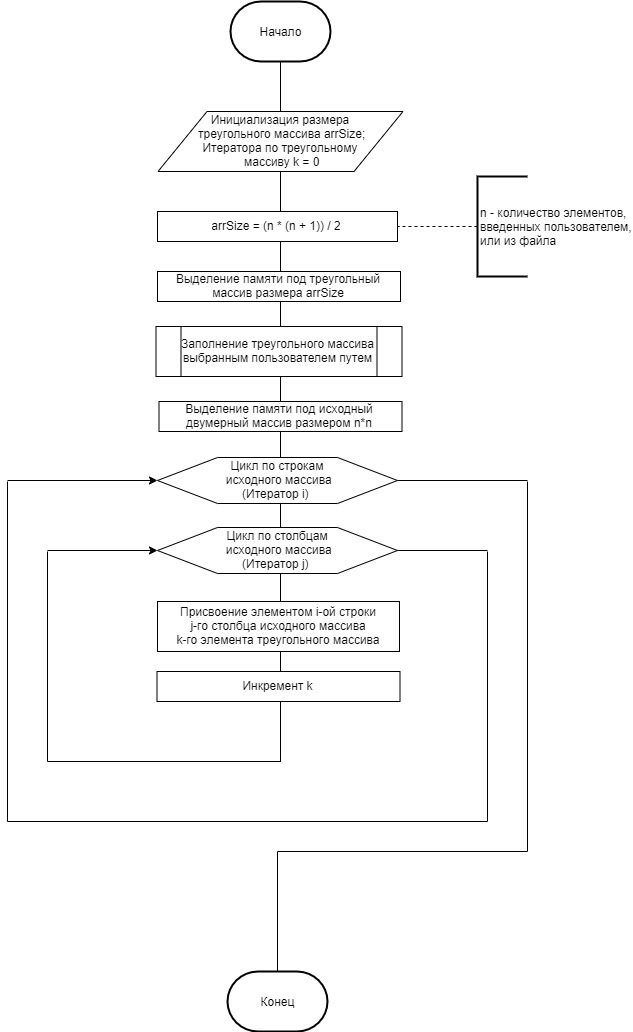


Рисунок 1 – Блок-схема восстановления исходной матрицы

Согласно постановке задачи, для составления программы будут использоваться алгоритмы, блок схема которых представлена выше.

# Форматы представления данных

Программа использует следующие переменные:

Таблица 1 – Переменные, используемые в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| n | int | Количество элементов |
| arrSize | int | Размер массива |
| running | bool | Определяет, запущена ли программа |
| menuItem | int | Для ввода пункта меню |
| i | int | Для цикла чередования строк |
| j | int | Для цикла чередования столбцов |
| file | ifstream | Занесение значений из файла в массив |
| filePath | string | Путь файла |
| triangleArr | int\* | Одномерный треугольный массив |
| valCorrect | bool | Проверка формата значения в файле |
| origArr | int\*\* | Массив |
| saveInFile | bool | Хочет ли пользователь записать массив в файл |
| color | int | Цвет вывода в консоль |
| fileOutput | ofstream | Вывод данных в файл |
| MenuItems | enum | Проверка пунктов меню |
| isPathValid | bool | Проверка пути файла |
| Colors | enum | Цвета |
| saveTriangle | bool | Определяет , хочет ли пользователь сохранить треугольную матрицу |
| saveOrig | bool | Определяет , хочет ли пользователь сохранить оригинальную матрицу |

Для задания максимального и минимального размера массивов, предела генерирования случайных чисел, а также ввода максимального пункта меню используются следующие константы:

Таблица 2 – Константы, используемы в программе

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Тип** | **Значение** | **Описание** |
| minArrSize | const int | 1 | Минимальное число элементов массива |
| INT\_MAX | const int | 2147483647 | Максимальное целое число |
| maxRandValue | const int | 100 | Максимальное случайно сгенерированное число |
| indent | const int | 2 | Длина отступа между элементами |

# Структура программы

В силу большого количества функций программа разделена на пять исполняемых модулей, из которых один является основным и отвечает за запуск программы, четыре оставшихся содержат в себе функции, необходимые для работы программы.

Модуль Source:

Таблица 3 – Функции, составляющие модуль Source

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| main | Запуск программы |
| Menu | Вывод главного меню, выбор пункта меню |

Модуль ArrInput:

Таблица 4 – Функции, составляющие модуль ArrInput

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| EndingOutput | Вывод треугольной и исходной матрицы в консоль и в файл |
| InputArrSize | Ввод количества элементов |
| ManualInput | Заполнение массива вручную |
| RandomInput | Заполнение массива случайными элементами |
| FilePathCheckReturnForInput | Проверка и возврат пути к файлу |
| FileInput | Проверка и заполнение массива из файла |

Модуль ArrChange:

Таблица 5 – Функции, составляющие модуль ArrChange

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| ArrChange | Восстановление исходной матрицы, выполнение задания |

Модуль ArrOutput:

Таблица 6 – Функции, составляющие модуль ArrOutput

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| SetColor | Изменение цвета текста, выводимого в консоль |
| CoutWithColor | Вывод в консоль текста с заданным цветом |
| NumberLenght | Подсчет длины числа |
| OutputInConsoleTriangleArr | Вывод в консоль треугольной матрицы |
| OutputInConsoleOrigArr | Вывод в консоль исходной матрицы |
| FilePathCheckReturnForOutput | Проверка и возврат пути файла |
| OutputInFile | Вывод обеих матриц в файл |

Модуль VarCheck:

Таблица 7 – Функции, составляющие модуль VarCheck

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| InputCheck | Проверка на правильность введённых данных задаваемого типа |
| GetInt | Проверка на правильность введённых данных типа int |
| GetBool | Проверка на правильность введённых данных типа bool |

# Описание хода выполнения лабораторной работы

* В ходе лабораторной работы было создано решение (Lab1) в интегрированной среде разработки Microsoft Visual Studio C++ 2019. В нём был создан проект.
* При работе программы с файлами нужно было добавить проверки на валидность имени файла, а также на то, создан ли файл или нет при сохранении.
* При получении пользовательского ввода необходимо было добавить проверку, на его соответствие предполагаемому типу данных и условиям выбора.
* Перед повторением программы необходимо очищать память для того, чтобы не возникало непредвиденных ошибок, связанных с заполнением областей памяти старыми числами.
* Программа после запуска выдавала одни и те же результаты, хотя в коде использовался вызов функции rand, возвращающей случайное число. После изучения справочной системы выяснилось, что необходимо использовать функцию srand для начальной инициализации генератора случайных чисел. После этого программа стала работать правильно.

# Результат работы программы

В результате работы программа выводит два массива различных цветов, первый массив является треугольной матрицей, а второй — результатом работы программы.

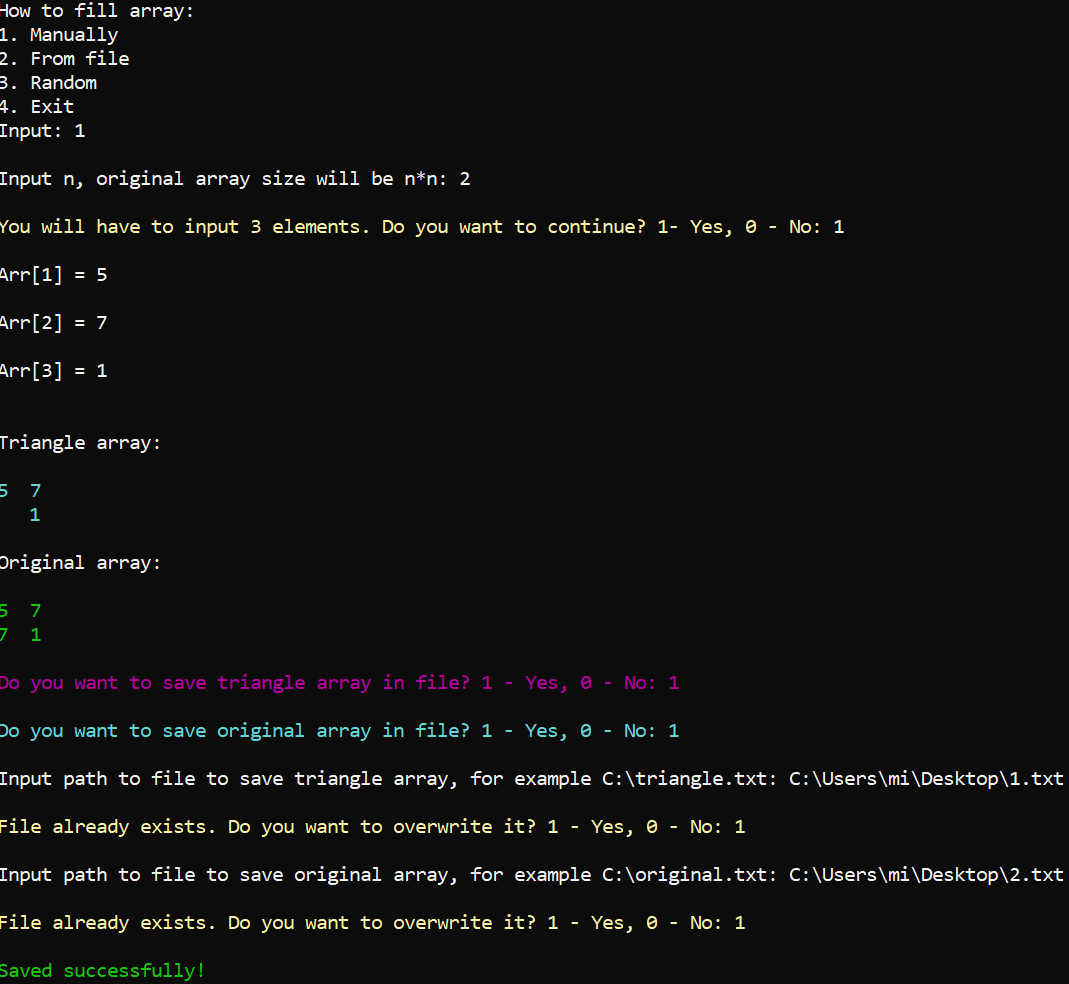


Рисунок 2 – Заполнение массива с клавиатуры, сохранение обеих матриц в файл

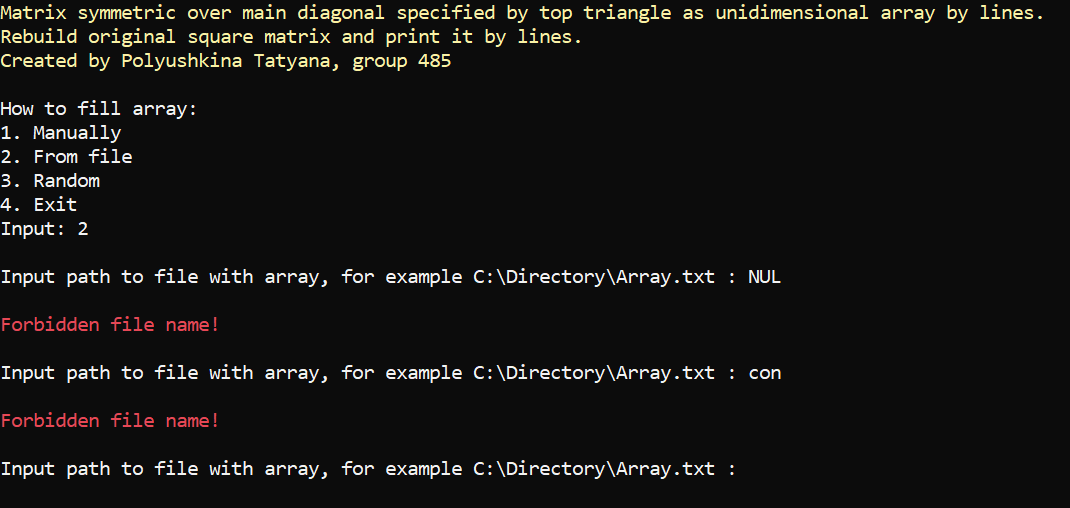


Рисунок 3 – Проверка на запрещенные имена

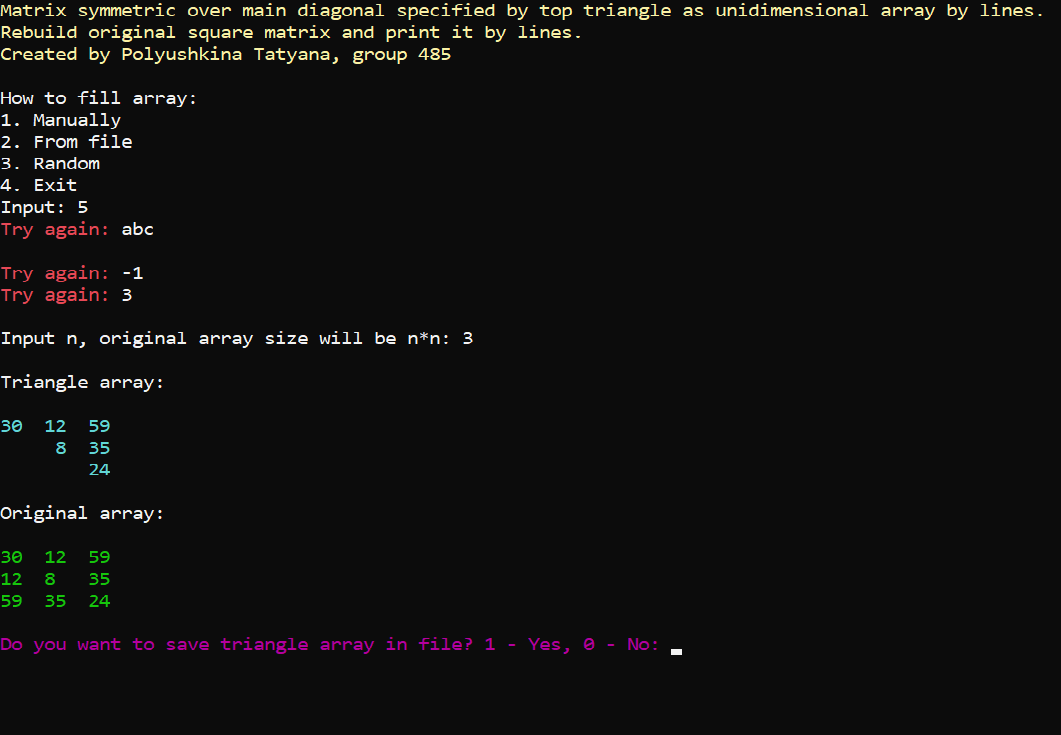


Рисунок 4 – Случайное заполнение массива, проверка правильности формата вводимых данных

# Текст программы

# [--- Начало программы ---]

**// ArrChange.h**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#pragma once

int\*\* ArrChange(int\* triangleArr, int n);

**// ArrInput.h**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#pragma once

void ManualInput();

void RandomInput();

void FileInput();

**// ArrOutput.h**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#pragma once

#include <string>

using namespace std;

enum { blue = 9, green, azure, red, purple, yellow, white };

int NumberLenght(int number);

void CoutWithColor(int color, string message);

void OutputInConsoleTriangleArr(int\* arr, int n, int maxNumberForSetw);

void OutputInConsoleOrigArr(int\*\* arr, int n, int maxNumberForSetw);

void OutputInFile(int\*\* origArr, int\* triangleArr, int n, int lenghtToSetw, bool saveTriangle, bool saveOrig);

void SetColor(int color);

**// VarCheck.h**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#pragma once

int GetInt();

bool GetBool();

**// Source.cpp**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include <iostream>

#include <Windows.h>

#include "VarCheck.h"

#include "ArrInput.h"

#include "ArrOutput.h"

using namespace std;

enum MenuItems {

minMenuItem = 0, manual, file, random, close, maxMenuItem

};

bool Menu() { // Вывод меню и проверка введённых данных

cout << "How to fill array: " << endl << "1. Manually" << endl << "2. From file" << endl << "3. Random" << endl <<

"4. Exit" << endl << "Input: ";

int menuItem; // пункт меню для ввода пользователем

bool running = true; // Для возврата в main, проверка запущена ли программа

while (true) { // ввод пользователем элемента меню и его проверка

menuItem = GetInt(); // ввод элемента меню, проверка на int

if (menuItem > minMenuItem && menuItem < maxMenuItem) { // проверка на выход за пределы меню

break;

}

else CoutWithColor(red, "Try again: ");

}

switch (menuItem) // соотношение данных, если введеные пользователем данные совпадают с условием, то выполняются определенные действия.

{

case manual:

ManualInput();

running = true;

break;

case file:

FileInput();

running = true;

break;

case random:

RandomInput();

running = true;

break;

case close:

running = false;

break;

}

return running; // если running = true - программа продолжается, если false, то программа завершается

}

int main() { // работа программы

CoutWithColor(yellow, "Matrix symmetric over main diagonal specified by top triangle as unidimensional array by lines.\nRebuild original square matrix and print it by lines.\nCreated by Polyushkina Tatyana, group 485\n\n");

bool running = true; // Для определения запущенности программы

while (running)

{

running = Menu();

}

return 0;

}

**// ArrChange.cpp**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

int\*\* ArrChange(int\* triangleArr, int n) { // создание и заполение исходного массива

int\*\* origArr = new int\* [n];// выделение памяти под строки исходного массива

for (int i = 0; i < n; ++i) { // выделение памяти под столбцы исходного массива

origArr[i] = new int[n];

}

int k = 0;

// заполнение исходной матрицы значениями из треугольной

for (int i = 0; i < n; ++i) { // итерация по строкам массива

for (int j = i; j < n; ++j, ++k) { // итерация по столбцам массива

// зеркалирование матрицы

origArr[i][j] = triangleArr[k];

origArr[j][i] = triangleArr[k];

}

}

return origArr; // возврат адреса исходной матрицы

}

**// ArrInput.cpp**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include <iostream>

#include <ctime>

#include <string>

#include <fstream>

#include <filesystem>

#include "VarCheck.h"

#include "ArrChange.h"

#include "ArrOutput.h"

using namespace std;

using namespace std::filesystem;

const int minArrSize = 1;

const int maxRandValue = 100;

const int indent = 2;

void EndingOutput(int\* triangleArr, int\*\* origArr, int n, int maxNumberForSetw) { // вывод в консоль и в файл обоих массивов

int lenghtToSetw = NumberLenght(maxNumberForSetw) + indent; // длина отступа

OutputInConsoleTriangleArr(triangleArr, n, lenghtToSetw); // вывод в консоль треугольной матрицы

OutputInConsoleOrigArr(origArr, n, lenghtToSetw); // вывод в консоль исходной матрицы

SetColor(purple);

cout << "Do you want to save triangle array in file? 1 - Yes, 0 - No: ";

bool saveTriangle = GetBool(); // запрос у пользователя, хочет ли он сохранить в файл оба массива

SetColor(azure);

cout << "\nDo you want to save original array in file? 1 - Yes, 0 - No: ";

bool saveOrig = GetBool();

SetColor(white);

if (saveTriangle || saveOrig)

OutputInFile(origArr, triangleArr, n, lenghtToSetw, saveTriangle, saveOrig);

else

cout << endl;

delete[] triangleArr; // освобождение памяти, выделенной под треугольный массив

for (int i = 0; i < n; ++i) { // освобождение памяти, выделенной под столбцы двумерного массива

delete[] origArr[i];

}

delete[] origArr; // освобождение памяти, выделенной под строки двумерного массива

}

int InputN() { //ввод количества элементов n

cout << "\nInput n, original array size will be n\*n: ";

int n;

while (true) { // ввод пользователем элемента n и его проверка

n = GetInt();

if (n > minArrSize) {

break;

}

else

CoutWithColor(red, "\nTry again: ");

}

return n;

}

void ManualInput() { // ввод вручную

int n; // количество элементов на ввод пользователю

int triangleArrSize; // размер треугольного массива по формуле (n \* (n + 1)) / 2

int maxNumberForSetw = 0; // Максимальное число, для вычисления длины отступа

bool userAgreed = false; // Хочет ли пользователь ввести количество элементов = triangleArrSize

while (!userAgreed) { // пока пользователь не согласен ввести количество элементов triangleArrSize

n = InputN(); // ввод n

triangleArrSize = (n \* (n + 1)) / 2; // вычисление triangleArrSize

string message = "\nYou will have to input ";

message += to\_string(triangleArrSize);

message += " elements. Do you want to continue? 1- Yes, 0 - No: ";

CoutWithColor(yellow, message);

userAgreed = GetBool();

}

int\* triangleArr = new int[triangleArrSize]; // создание одномерного треугольного массива размерностью triangleArrSize

for (int i = 0; i < triangleArrSize; ++i) { // присваевание элементам массива введенные пользователем значения

cout << endl << "Arr[" << i + 1 << "] = ";

triangleArr[i] = GetInt();

if (abs(triangleArr[i]) > maxNumberForSetw) { // поиск максимального по модулю элемента для вычисления отступа

maxNumberForSetw = abs(triangleArr[i]);

}

}

cout << endl;

int\*\* origArr = ArrChange(triangleArr, n); // объявление двумерного исходного массива

EndingOutput(triangleArr, origArr, n, maxNumberForSetw); // вывод в консоль и в файл обоих массивов

}

void RandomInput() { // генерация случайных чисел для массива

srand(static\_cast<unsigned int>(time(nullptr))); // для установки начала последовательности генерируемой rand()

int n = InputN();// ввод n

int triangleArrSize = (n \* (n + 1)) / 2; // размер треугольного массива по формуле (n \* (n + 1)) / 2

int maxNumberForSetw = 0; // Максимальное число, для вычисления длины отступа

int\* triangleArr = new int[triangleArrSize]; // создание одномерного треугольного массива размерностью triangleArrSize

for (int i = 0; i < triangleArrSize; ++i) { // помещение рандомно сгенерированных чисел в массив

triangleArr[i] = rand() % maxRandValue; // ограничение рандомного числа

if (abs(triangleArr[i]) > maxNumberForSetw) { // поиск максимального по модулю элемента для вычисления отступа

maxNumberForSetw = triangleArr[i];

}

}

int\*\* origArr = ArrChange(triangleArr, n); // объявление двумерного исходного массива

EndingOutput(triangleArr, origArr, n, maxNumberForSetw); // вывод в консоль и в файл обоих массивов

}

string FilePathCheckReturnForInput() { // проверка пути для чтения из файла

string filePath;

bool isPathValid = false;

while (!isPathValid) { // while isPathValid == false, проверка валидности введенного пути

cout << "\nInput path to file with array, for example C:\\Directory\\Array.txt : ";

cin >> filePath;

cin.ignore(INT\_MAX, '\n');

if (!ifstream(filePath)) { // существует ли файл

CoutWithColor(red, "\nFile does not exist!\n");

continue;

}

try {

is\_regular\_file(filePath);

}

catch (...) {

CoutWithColor(red, "\nForbidden file name!\n");

continue;

}

ifstream file(filePath);

if (!file) { //проверка доступа к файлу

CoutWithColor(red, "\nAccess denied!\n");

continue;

}

isPathValid = true;

}

return filePath;

}

void FileInput() { //считывание данных из файла

bool valCorrect = false; // проверка на соответствие int и выхода из цикла while, если все значения корректы

int n;

int maxNumberForSetw = 0;

int\* triangleArr; // объявление указателя на треугольную матрицу

while (!valCorrect) {

string filePath = FilePathCheckReturnForInput();

ifstream file(filePath);

if (!(file >> n) || n < minArrSize) { //проверка размера массива

CoutWithColor(red, "\nIncorrect array size.\n");

continue;

}

int triangleArrSize = (n \* (n + 1)) / 2; // размер треугольного массива по формуле (n \* (n + 1)) / 2

triangleArr = new int[triangleArrSize]; // создание одномерного треугольного массива размерностью triangleArrSize

bool toContinue = false; // для перехода в начало цикла while, так как continue продолжит цикл for

for (int i = 0; i < triangleArrSize; ++i) { // проверка элементов массива

if (!(file >> triangleArr[i])) {

CoutWithColor(red, "\nIncorrect array value.\n");

toContinue = true;

break;

}

if (abs(triangleArr[i]) > maxNumberForSetw) {

maxNumberForSetw = abs(triangleArr[i]);

}

}

if (toContinue) continue;

valCorrect = true;

int\*\* origArr = ArrChange(triangleArr, n); // объявление двумерного исходного массива

EndingOutput(triangleArr, origArr, n, maxNumberForSetw); // вывод в консоль и в файл обоих массивов

}

}

**// ArrOutput.cpp**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include <iostream>

#include <fstream>

#include <Windows.h>

#include <string>

#include <filesystem>

#include "VarCheck.h"

using namespace std;

using namespace std::filesystem;

enum Colors { blue = 9, green, azure, red, purple, yellow, white };

const int indent = 2;

void SetColor(int color) { // изменение цвета в консоли

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color);

}

void CoutWithColor(int color, string message) { // вывод сообщения message с цветом color

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color); // изменение цвета на color

cout << message; // вывод сообщения

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), white); // изменение цвета на белый

}

int NumberLenght(int number) { // вычисление длины числа

int lenght = 0;

while (number != 0) {

++lenght;

number /= 10;

}

return lenght;

}

void OutputInConsoleOrigArr(int\*\* arr, int n, int lenghtToSetw) {

cout << "Original array:\n\n";

SetColor(green);

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

for (int j = 0; j < n; ++j) {

cout << arr[i][j]<< setw(lenghtToSetw);

}

cout << endl << left;

}

cout << endl << resetiosflags(ios::adjustfield);

SetColor(white);

}

void OutputInConsoleTriangleArr(int\* arr, int n, int lenghtToSetw) {

cout << "\nTriangle array:\n\n";

SetColor(azure);

int k = 0;

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

for (int j = 0; j < n - i; ++j, ++k) {

cout << arr[k] << setw(lenghtToSetw); // вывод элемента и задание отступа для следующего элемента

}

cout << endl;

int forTriangleOutput = lenghtToSetw \* (i + 1) + 2; // вычисление длины отступа слева

cout << setw(forTriangleOutput);

}

cout << setw(0) << endl;

SetColor(white);

}

string FilePathCheckReturnForOutput(string message) { // проверка пути для сохранения в файл

string filePath;

bool isPathValid = false;

while (!isPathValid) { // while isPathValid == false, проверка валидности введенного пути

cout << "\n"<< message;

cin >> filePath;

cin.ignore(INT\_MAX, '\n');

if (ifstream(filePath)) { // существует ли файл

try {

if (is\_regular\_file(filePath)) { // проверка на запрещенные имена (aux, con..)

SetColor(yellow);

cout << "\nFile already exists. Do you want to overwrite it? 1 - Yes, 0 - No: ";

bool toOverwrite = GetBool();

SetColor(white);

if (!toOverwrite) {

continue;

}

}

}

catch (...) {

CoutWithColor(red, "\nForbidden file name!\n");

continue;

}

}

ofstream file(filePath); // создание файла по заданному пути

if (!file) { // проверка на доступ к созданию файла

CoutWithColor(red, "\nAccess denied!\n");

continue;

}

isPathValid = true;

}

return filePath; // возврат корректного пути

}

void OutputInFile(int\*\* origArr, int\* triangleArr, int n, int lenghtToSetw, bool saveTriangle, bool saveOrig) {

if (saveTriangle) {

ofstream file(FilePathCheckReturnForOutput("Input path to file to save triangle array, for example C:\\triangle.txt: "));

file << n << endl;

int k = 0;

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

for (int j = 0; j < n - i; ++j, ++k) {

file << triangleArr[k] << endl;

}

}

}

if (saveOrig) {

ofstream file(FilePathCheckReturnForOutput("Input path to file to save original array, for example C:\\original.txt: "));

file << n << endl << n << endl;

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

for (int j = 0; j < n; ++j) {

file << origArr[i][j] << endl;

}

}

}

CoutWithColor(green, "\nSaved successfully!\n\n");

}

**// VarCheck.cpp**

**// Лабораторная работа №1.**

**// Студент группы 485, Полюшкина Татьяна Юрьевна. 2019 год**

#include <iostream>

#include "ArrOutput.h"

using namespace std;

template <typename T> // создание шаблона T

T InputCheck() { // проверка ввода пользователя

T userInput; // создание переменной заданного типа для записи ввода пользователя

cin >> userInput;

while (cin.fail()) { // пока введенные данные не соответствуют типу

CoutWithColor(red, "\nTry again: ");

cin.clear(); // очищает cin.fail()

cin.ignore(INT\_MAX, '\n'); // очистка буфера

cin >> userInput;

}

cin.ignore(INT\_MAX, '\n');

return userInput;

}

int GetInt() { // ввод пользователем, проверка и возврат значения int

return InputCheck<int>();

}

bool GetBool() { // ввод пользователем, проверка и возврат значения bool

return InputCheck<bool>();

}

**[--- Конец программы ---]**