**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Информационных систем**

отчет

**по лабораторной работе**

**по дисциплине «Программирование»**

Тема: **добавить инфраструктуру переключения между практическими заданиями.**

Студент гр.0324 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кошеляев А.С

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Глущенко А.Г

Санкт-Петербург

2020

### Цель работы.

Разработать алгоритм и написать программу на языке С++, объединяющий инфраструктуру переключения между практическими заданиями.

### Основные теоретические положения.

Использование функций позволяет:

1) значительно упростить разработку сложных программ;

2) сократить объем текста программы и генерируемого результирующего кода программы;

3) значительно упростить отладку и модификацию программ;

4) распределить работу над одной программой между различными исполнителями программистами.

Фактически разработка более-менее сложных программ практически невозможна без использования функций.

Инструкцию безусловного перехода **goto,** инструкция goto обеспечивает переход на выполнение инструкции отмеченной с помощью **метки**.

Формат записи: goto **<Метка>;**

Метка представляет собой некоторый идентификатор, за которым следует символ’**:** ’. Меткой может быть помечена любая инструкция, находящаяся в той же функции, в которой находится оператор **goto**.

Оператор if-else управляет условным ветвлением. Операторы в *if-branch* выполняются, только если *condition* результатом вычисления является ненулевое значение (или **true**). Если значение *condition* не равно нулю, выполняется следующая инструкция, а инструкция, следующая за необязательным, **else** пропускается. В противном случае пропускается Следующая инструкция, и, если имеется **else** оператор после оператора, **else** выполняется инструкция.

### Постановка задачи.

Необходимо объединить все 4 практические работы в единый проект. Нужно добавить инфраструктуру переключения между заданиями (интерактивное меню). 

#### **Выполнение работы.**

Выполняя лабораторную работу опираясь на полученные знания в практических работах, получаю интерактивное меню рис1.

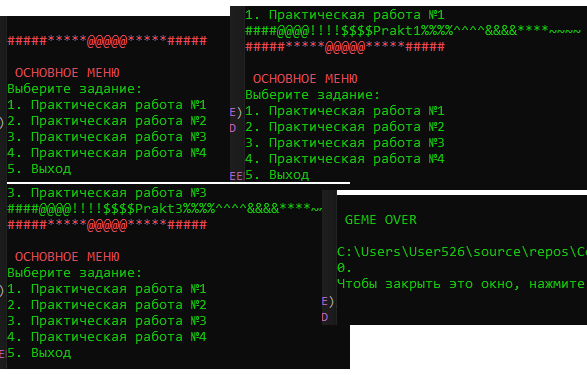


Рисунок 1 Интерактивное меню.

Размещение кода практических работ в рабочем интерактивном меню приведён в приложение А.

### Выводы.

Данная лабораторная работа позволила объединить полученные знания, и позволила получить дополнительный опыт в написании кода С++.

### **Приложение А**

### **Полный код программы**

//Лабораторка добавить инфраструктуру переключения между практическими заданиями

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <chrono>

#include <string>

#include <fstream>

#include <algorithm>

#include <ctime>

#include <iomanip>

#include <windows.h>

using namespace std;

int menu(int xz) {

int qwe = 0;

HANDLE O = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(O, FOREGROUND\_RED | FOREGROUND\_INTENSITY);

cout << "\n#####\*\*\*\*\*@@@@@\*\*\*\*\*#####\n";

cout << "\n ОСНОВНОЕ МЕНЮ\n";

SetConsoleTextAttribute(O, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << setw(4) << "Выберите задание:" << "\n";

cout << setw(4) << "1. Практическая работа №1\n";

cout << setw(4) << "2. Практическая работа №2\n";

cout << setw(4) << "3. Практическая работа №3\n";

cout << setw(4) << "4. Практическая работа №4\n";

cout << setw(4) << "5. Выход\n";

cin >> qwe;

cin.sync();

system("cls");

return qwe;

}

void print(int A[6][6], int n) {

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

{

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

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

cout << endl;

}

}

void random(int A[6][6], int x) {

int \*end = A[0] + x \* x - 1; //указатель на последний элемент массива

srand(time(0));

for (int \*ptr = A[0], i = 1; ptr <= end; ptr++, i++)

\*ptr = rand() % (x\*x) + 1;

}

void calculator(int A[6][6], int p) {

int x = 0, y = 0;

int \*end = A[0] + p \* p - 1;

cout << "\n Vvidite celoe chislo \n";

cin >> x;

cin.sync();

int \*vi = &x;

cout << "\n Vibraty operacu " << "\n 0 vichetanie\n 1 slogenie\n 2 umnogenie\n 3 delenie\n";

cin >> y;

cin.sync();

switch (y)

{

case 0:

for (int \*ptr = A[0], i = 1; ptr <= end; ptr++, i++)

\*ptr -= \*vi;

break; cout << "\n ERROR\n";

case 1: {

for (int \*ptr = A[0], i = 1; ptr <= end; ptr++, i++)

\*ptr += \*vi; }

break; cout << "\n ERROR\n";

case 2: {

for (int \*ptr = A[0], i = 1; ptr <= end; ptr++, i++)

\*ptr \*= \*vi; }

break; cout << "\n ERROR\n";

case 3: {

for (int \*ptr = A[0], i = 1; ptr <= end; ptr++, i++)

\*ptr /= \*vi; }

break; cout << "\n ERROR\n";

}

}

void VIVOD(int A[6][6]) {

int u = 0;

switch (u)

{

case 0: for (int i = 0; i <= 5; i++) {

cout << setw(4) << A[i][0] << "\n";

Sleep(500); u++;

}

case 1:

{system("cls");

cout << setw(4) << A[0][0] << "\n";

cout << setw(4) << A[1][0] << "\n";

cout << setw(4) << A[2][0] << "\n";

cout << setw(4) << A[3][0] << "\n";

cout << setw(4) << A[4][0] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 2:

{system("cls");

cout << setw(4) << A[0][0] << "\n";

cout << setw(4) << A[1][0] << "\n";

cout << setw(4) << A[2][0] << "\n";

cout << setw(4) << A[3][0] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 3:

{system("cls");

cout << setw(4) << A[0][0] << "\n";

cout << setw(4) << A[1][0] << "\n";

cout << setw(4) << A[2][0] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 4:

{system("cls");

cout << " " << A[0][0] << "\n";

cout << " " << A[1][0] << "\n";

cout << " " << A[2][0] << "\t" << A[2][1] << "\n";

cout << " " << A[3][0] << "\t" << A[3][1] << "\n";

cout << " " << A[4][0] << "\t" << A[4][1] << "\n";

cout << " " << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 5:

{system("cls");

cout << " " << A[0][0] << "\n";

cout << " " << A[1][0] << "\t" << A[1][1] << "\n";

cout << " " << A[2][0] << "\t" << A[2][1] << "\n";

cout << " " << A[3][0] << "\t" << A[3][1] << "\n";

cout << " " << A[4][0] << "\t" << A[4][1] << "\n";

cout << " " << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 6:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 7:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 8:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 9:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 10:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 11:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\n";

Sleep(500); u++; }

case 12:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\n";

Sleep(500); u++; }

case 13:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 14:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 15:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 16:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 17:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 18:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 19:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 20:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 21:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 22:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 23:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\n";

Sleep(500); u++; }

case 24:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\t" << A[5][4] << "\n";

Sleep(500); u++; }

case 25:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\t" << A[5][4] << "\t" << A[5][5] << "\n";

Sleep(500); u++; }

case 26:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\t" << A[4][5] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\t" << A[5][4] << "\t" << A[5][5] << "\n";

Sleep(500); u++; }

case 27:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\t" << A[3][5] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\t" << A[4][5] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\t" << A[5][4] << "\t" << A[5][5] << "\n";

Sleep(500); u++; }

case 28:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\t" << A[2][5] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\t" << A[3][5] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\t" << A[4][5] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\t" << A[5][4] << "\t" << A[5][5] << "\n";

Sleep(500); u++; }

case 29:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\t" << A[1][5] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\t" << A[2][5] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\t" << A[3][5] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\t" << A[4][5] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\t" << A[5][4] << "\t" << A[5][5] << "\n";

Sleep(500); u++; }

case 30:

{system("cls");

cout << setw(4) << A[0][0] << "\t" << A[0][1] << "\t" << A[0][2] << "\t" << A[0][3] << "\t" << A[0][4] << "\t" << A[0][5] << "\n";

cout << setw(4) << A[1][0] << "\t" << A[1][1] << "\t" << A[1][2] << "\t" << A[1][3] << "\t" << A[1][4] << "\t" << A[1][5] << "\n";

cout << setw(4) << A[2][0] << "\t" << A[2][1] << "\t" << A[2][2] << "\t" << A[2][3] << "\t" << A[2][4] << "\t" << A[2][5] << "\n";

cout << setw(4) << A[3][0] << "\t" << A[3][1] << "\t" << A[3][2] << "\t" << A[3][3] << "\t" << A[3][4] << "\t" << A[3][5] << "\n";

cout << setw(4) << A[4][0] << "\t" << A[4][1] << "\t" << A[4][2] << "\t" << A[4][3] << "\t" << A[4][4] << "\t" << A[4][5] << "\n";

cout << setw(4) << A[5][0] << "\t" << A[5][1] << "\t" << A[5][2] << "\t" << A[5][3] << "\t" << A[5][4] << "\t" << A[5][5] << "\n";

Sleep(500); }

}

}

void Sortirovka(int A[6][6], int n) {

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

{

for (int j = (n\*n) - 1; j > i; j--)

{

if (((int \*)A)[j] < ((int \*)A)[j - 1])

{

int temp = ((int \*)A)[j];

((int \*)A)[j] = ((int \*)A)[j - 1];

((int \*)A)[j - 1] = temp;

}

}

}

}

void Removing(char \*strs) {

int t = 0;

string str = string(strs);

for (int i = 0; i < size(str); i++) { // удаление повторных символов

t = 0;

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

if (!isalpha((unsigned char)str[j])) {

if (str[j] == str[j + 1]) {

if (str[j] == '.') {

t++;

if (t >= 3)

str.erase(j, 1);

}

else

str.erase(j, 1);

}

}

}

}

strcpy(strs, str.c\_str());

cout << str << endl << endl;

setlocale(0, "RU");

}

void Register(char \* str) {

for (int i = 2; i < strlen(str); i++) // выравниваем регистр

{

str[0] = toupper(str[0]);

if (str[i - 2] == '.' || str[i - 1] == '.' || str[i - 2] == '?' || str[i - 1] == '?') {

str[i] = toupper(str[i]);

}

else {

if (str[i - 1] != ' ')

str[i] = tolower(str[i]);

else

str[i] = str[i];

}

}

cout << str << endl << endl;

setlocale(0, "RU");

}

void Task(char pch[500]) { // знаю криво получилось только так прошу при оценке написать свой прмер

char \*pchh = new char(500);

pchh = strtok(pch, " ,.-"); // самая большая проблемма в этойй строчке

while (pchh != NULL) // пока есть лексемы

{

string str = string(pchh);

for (int i = 1; i < size(str); i++) {

if (str[0] == str[i])

cout << pchh << "\n";

}

pchh = strtok(NULL, " ,.-)\*");

}

cout << endl;

setlocale(0, "RU");

}

void Number(char \*ku)

{

string kuk = string(ku);

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

replace(kuk.begin(), kuk.end(), '1', 'a');

replace(kuk.begin(), kuk.end(), '2', 'b');

replace(kuk.begin(), kuk.end(), '3', 'c');

replace(kuk.begin(), kuk.end(), '4', 'd');

replace(kuk.begin(), kuk.end(), '5', 'e');

replace(kuk.begin(), kuk.end(), '6', 'f');

replace(kuk.begin(), kuk.end(), '7', 'g');

replace(kuk.begin(), kuk.end(), '8', 'h');

replace(kuk.begin(), kuk.end(), '9', 'i');

replace(kuk.begin(), kuk.end(), '0', 'z');

}

cout << kuk << endl << endl;

setlocale(0, "RU");

}

bool ReadFile(const char \*FileName)

{

char yes[500];

char copy[500];

char \*S;

ifstream File;

File.open(FileName);

if (!File.is\_open()) // Проверили удалось ли открыть файл

{

cout << "Открыть файл не удалось! \n";

return 0;

}

while (!File.eof()) // Читаем все строки из файла и выводим их на экран

File.getline(yes, 500);

S = yes;

cout << "Original:\n" << S << endl << endl;

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nУдаление лишних знаков\n";

Removing(S); // удаление повторных символов

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nИсправление регистра букв\n";

Register(S); // выравниваем регистр

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nЗадание 3 вариант 5\n";

strcpy(copy, S);

Task(copy); // выводим слова, в которых есть буквы такие же как и первая буква слова

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nЗадание 4 вариант 4\n";

Number(S);

cout << "\n GAME OVER\n";

File.close(); // Закрываем файл

return 1;

}

void WriteText() {

cin.get();

char yes[500];

char copy[500];

char \*S;

cout << "Введите текст содержащий от 1 до 50 слов, в каждом из которых от 1 до 10 строчных латинских букв и цифр. Между соседними словами произвольное количество пробелов. За последним символом стоит точка. \n";

cin.getline(yes, 500);

S = yes;

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nУдаление лишних знаков\n";

setlocale(0, ".866");

Removing(S); // удаление повторных символов

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nИсправление регистра букв\n";

setlocale(0, ".866");

Register(S); // выравниваем регистр

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nЗадание 3 вариант 5\n";

strcpy(copy, S);

setlocale(0, ".866");

Task(copy); // выводим слова, в которых есть буквы такие же как и первая буква слова

cout << "\n###\*\*\*@@@\*\*\*###\n";

cout << "\nЗадание 4 вариант 4\n";

setlocale(0, ".866");

Number(S);

cout << "\n GAME OVER\n";

}

int main()

{

Mem1:

int xz = 0;

setlocale(LC\_ALL, "Russian");

int lab = menu(xz);

if (lab == 1) {

cout << setw(4) << "1. Практическая работа №1\n";

cout << "int " << sizeof(int) << "\t short int " << sizeof(short int)

<< "\t long int " << sizeof(long int) << "\t float " << sizeof(float)

<< "\t double " << sizeof(double) << "\t long double " << sizeof(long double) << "\t char " << sizeof(char) << "\t bool " << sizeof(bool) << "\n";

int a = 65535;

int z = 1 << 31;

for (int i = 0; i < 32; i++)

{

if (a&z)

cout << 1;

else

cout << 0;

a <<= 1;

if (i == 0)

cout << "\t ";

}

cout << "\n";

union {

float b = 2.718;

int c;

}x;

for (int i = 0; i < 32; i++)

{

if (z&x.c)

cout << 1;

else

cout << 0;

x.c <<= 1;

if (i == 0)

cout << "\t ";

if (i == 8)

cout << "\t";

}

cout << "\n";

goto Mem1;

}

if (lab == 2) {

cout << setw(4) << "2. Практическая работа №2\n";

const int a = 100;

short A[a];

M1:

auto start = std::chrono::system\_clock::now();

int q, max = A[0], min = A[0], ser = 0, o = 0, n;

auto srand(time(0));

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

A[i] = rand() % (199 - 1) - 99;

auto end = std::chrono::system\_clock::now();

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

cout << "A[" << i << "]=" << A[i] << "\n";

std::chrono::duration<double> diff = end - start;

cout << "\n" << "Time: " << diff.count() << "s";

cout << "\n" << "Create data array: 1 " << "\n" << "Start sorting: 2" << "\n" << "Start searching for max and min element: 3" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

default: cout << "ERROR \n";

}

M2:

start = chrono::system\_clock::now();

for (int i = 1; i < a; i++)

{

int x = A[i];

int j = i;

while ((j > 0) && (x < A[j - 1]))

{

A[j] = A[j - 1];

j--;

}

A[j] = x;

}

end = chrono::system\_clock::now();

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

{

cout << setw(5) << left << A[i];

}

chrono::duration<double> set = end - start;

cout << "\n" << "Time: " << set.count() << "s";

cout << "\n" << "Create data array: 1 " << "\n" << "Start sorting: 2" << "\n" << "Start searching for max and min element: 3" << "\n Mean: 4" << "\n Item: 5" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

case 4: goto M4;

case 5: goto M5;

default: cout << "ERROR \n";

}

M3:

start = chrono::system\_clock::now();

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

if (A[i] > max)

max = A[i];

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

if (A[i] < min)

min = A[i];

end = chrono::system\_clock::now();

cout << "Min: " << min << "\n" << "Max: " << max << "\n";

std::chrono::duration<double> tide = end - start;

cout << "\n" << "Time sorted data array: " << tide.count() << "s";

cout << "\n" << "\n Create data array: 1 " << "\n Start sorting: 2" << "\n Start searching for max and min element: 3" << "\n Mean: 4" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

case 4: goto M4;

default: cout << "ERROR \n";

}

M4:

ser = (max + min) / 2;

cout << "\n" << ser << " Mean" << "\n";

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

{

if (A[i] == ser)

{

cout << "A[" << i << "]" << " number " << "\n";

++o;

}

}

cout << "number of array elements = " << ser << "= " << o;

cout << "\n" << "\n Create data array: 1 " << "\n Start sorting: 2" << "\n Start searching for max and min element: 3" << "\n Mean: 4" << "\n Next: 5" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

case 4: goto M4;

case 5: goto M5;

default: cout << "ERROR \n";

}

M5:

int e = 0;

cout << "\n Enter the number ";

cin >> n;

cin.sync();

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

{

if (A[i] < n)

{

++e;

}

}

cout << "The number of elements is less " << n << "= " << e;

cout << "\n" << "\n Create data array: 1 " << "\n Start sorting: 2" << "\n Start searching for max and min element: 3" << "\n Mean: 4" << "\n Back: 5" << "\n Next: 6" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

case 4: goto M4;

case 5: goto M5;

case 6: goto M6;

default: cout << "ERROR \n";

}

M6:

int p = 0;

cout << "\n Enter the number ";

cin >> n;

cin.sync();

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

{

if (A[i] > n)

{

++p;

}

}

cout << "The number of elements is greater " << n << "= " << p;

cout << "\n" << "\n Create data array: 1 " << "\n Start sorting: 2" << "\n Start searching for max and min element: 3" << "\n Mean: 4" << "\n Item: 5" << "\n Repeat: 6" << "\n Next: 7" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

case 4: goto M4;

case 5: goto M5;

case 6: goto M6;

case 7: goto M7;

default: cout << "ERROR \n";

}

M7:

cout << "\n Enter the number ";

cin >> n;

cin.sync();

int h = 0;

int s = -1;

int l = a;

start = chrono::system\_clock::now();

while (h <= l)

{

int mid = (h + l) / 2;

if (n == A[mid])

{

s = mid;

break;

}

if (n < A[mid])

l = mid - 1;

else

h = mid + 1;

}

end = chrono::system\_clock::now();

if (s == -1)

cout << "The element not found!\n";

else

cout << "Yes\n";

std::chrono::duration<double> xz = end - start;

cout << "\n" << "Time sorted data array: " << xz.count() << "s";

start = chrono::system\_clock::now();

int r = 0;

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

{

if (A[i] == n)

{

++r;

}

}

end = chrono::system\_clock::now();

cout << "\n Search by brute force " << n << " coincidences = " << r;

std::chrono::duration<double> acdc = end - start;

cout << "\n" << "Time sorted data array: " << acdc.count() << "s";

cout << "\n Difference:" << "Brute force -" << " binary search " << acdc.count() - xz.count() << " s";

cout << "\n" << "\n Create data array: 1 " << "\n Start sorting: 2" << "\n Start searching for max and min element: 3" << "\n Mean: 4" << "\n Item: 5" << "\n Item: 6" << "\n Repeat: 7" << "\n Next: 8" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

case 4: goto M4;

case 5: goto M5;

case 6: goto M6;

case 7: goto M7;

case 8: goto M8;

default: cout << "ERROR \n";

}

M8:

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

cout << "A[" << i << "]=" << A[i] << " WAS" << "\n";

int w;

cout << "Write 0 < A & B < 99 : \n";

cin >> n >> w;

cin.sync();

start = chrono::system\_clock::now();

int tmp;

tmp = A[n];

A[n] = A[w];

A[w] = tmp;

end = chrono::system\_clock::now();

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

cout << "A[" << i << "]=" << A[i] << " Became" << "\n";

std::chrono::duration<double> yo = end - start;

cout << "\n" << "Time sorted data array: " << yo.count() << "s";

cout << "\n" << "\n Create data array: 1 " << "\n Start sorting: 2" << "\n Start searching for max and min element: 3" << "\n Mean: 4" << "\n Item: 5 " << "\n Item: 6" << "\n Item: 7" << "\n Repeat: 8" << "\n";

cin >> q;

cin.sync();

switch (q)

{

case 1: goto M1;

case 2: goto M2;

case 3: goto M3;

case 4: goto M4;

case 5: goto M5;

case 6: goto M6;

case 7: goto M7;

case 8: goto M8;

default: cout << "\n the project is over \n";

}

goto Mem1;

}

if (lab == 3) {

cout << setw(4) << "3. Практическая работа №3\n";

const int n = 6;

const int m = 6;

int t = n / 2;

int A[n][m], B1[n][m], B2[n][m], B3[n][m], B4[n][m], C[n][m], tmp, next;

random(A, n);

VIVOD(A);

cout << "\t Original \n"; //Вывод оригинальной матрицы.

print(A, n);

for (int i = 0; i < n; i++) {//Делю оригинальный двумерный массив на 4 части. B1 [0-2][0-2] B2[0-2][3-5]

for (int j = 0; j < n; j++) { // B3[3-5][3-5] B4[3-5][0-2]

if (i < t && j < t)

B1[i][j] = A[i][j];

else

if (i < t && j >= t)

B2[i][j - t] = A[i][j];

else

if (i >= t && j >= t)

B3[i - t][j - t] = A[i][j];

else

if (i >= t && j < t)

B4[i - t][j] = A[i][j];

}

}

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

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

if (i < t && j < t)

C[i][j] = B4[i][j];

else

if (i < t && j >= t)

C[i][j] = B1[i][j - t];

else

if (i >= t && j >= t)

C[i][j] = B2[i - t][j - t];

else

if (i >= t && j < t)

C[i][j] = B3[i - t][j];

}

}

cout << "\t Option a\n";

print(C, n);

for (int i = 0; i < n / 2; i++) //меняю местами блоки крест на крест "просто решил попробовать и получилось" другим способом,

for (int j = 0; j < n / 2; j++) //в сторчках кода короче но для 3х заданий запарно.

{

tmp = A[i][j];

A[i][j] = A[i + (n + 1) / 2][j + (n + 1) / 2];

A[i + (n + 1) / 2][j + (n + 1) / 2] = tmp;

}

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

for (int j = (n + 1) / 2; j < n; j++)

{

tmp = A[i][j];

A[i][j] = A[i + (n + 1) / 2][j - (n + 1) / 2];

A[i + (n + 1) / 2][j - (n + 1) / 2] = tmp;

}

cout << "\t Option b\n";

print(A, n);

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

for (int j = 0; j < n; j++) { //Это задание переставить блоки С

if (i < t && j < t) // имея массивы B1,B2,B3,B4 переставлять легко.

C[i][j] = B4[i][j];

else

if (i < t && j >= t)

C[i][j] = B3[i][j - t];

else

if (i >= t && j >= t)

C[i][j] = B2[i - t][j - t];

else

if (i >= t && j < t)

C[i][j] = B1[i - t][j];

}

}

cout << "\t Option c\n";

print(C, n);

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

for (int j = 0; j < n; j++) { //Это задание переставить блоки D

if (i < t && j < t)

C[i][j] = B2[i][j];

else

if (i < t && j >= t)

C[i][j] = B1[i][j - t];

else

if (i >= t && j >= t)

C[i][j] = B4[i - t][j - t];

else

if (i >= t && j < t)

C[i][j] = B3[i - t][j];

}

}

cout << "\t Option d\n";

print(C, n);

Sortirovka(A, n);

cout << "\t Sortirovka\n";

print(A, n);

My1:

calculator(A, n);

cout << "\n Itogo\n";

print(A, n);

cout << "\n next calculator 4\n Выход в Основное меню 5\n";

cin >> next;

cin.sync();

switch (next)

{

case 4: goto My1;

case 5: {"\n Выход в Основное меню\n";

goto Mem1;

}

break; cout << "\n ERROR\n";

}

goto Mem1;

}

if (lab == 4) {

cout << setw(4) << "4. Практическая работа №4\n";

setlocale(0, "RU");

int vibor;

cout << "Selecting a line of text:\n1 - Manual input\n2 - File\n";

cin >> vibor;

if (vibor == 1) {

WriteText();

}

else if (vibor == 2) {

ReadFile("text.txt");

}

goto Mem1;

}

if (lab == 5) {

cout << "\n GEME OVER \n";

return 0;

}

return 0;

}