**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по курсовой работе**

**по дисциплине «Программирование»**

Тема: **Многофайловая программа и интерактивное меню.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4372 |  | Максимов Е. В. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2024

**Цель работы.**

Разработка программы для объединения в едином пользовательском интерфейсе ранее разработанных практических работ.

**Основные теоретические положения.**

**1. Многофайловые програмы**

Многофайловые программы в C++ представляют собой структуру, в которой код разбивается на несколько файлов. Это позволяет организовать код более эффективно, улучшает читаемость и упрощает его поддержку.

Основные компоненты многофайловой программы включают:

* Исходные файлы (.cpp), которые содержат реализацию функций и методов;
* Заголовочные файлы (.h или .hpp), которые содержат объявления функций, классов, структур и других элементов, которые могут использоваться в нескольких исходных файлах.

**2. Заголовочные файлы**

Заголовочные файлы играют ключевую роль в многофайловых программах. Они позволяют:

* Объявлять интерфейсы, так как заголовочные файлы содержат объявления функций и классов, что позволяет другим файлам использовать эти функции и классы без необходимости знать их реализацию.
* Избежать дублирования кода, потому что использование заголовочных файлов позволяет избежать повторного объявления одних и тех же функций и классов в разных исходных файлах.
* Упрощать управление зависимостями: заголовочные файлы могут включать другие заголовочные файлы, что позволяет организовать зависимости между различными частями программы.

**3. Директивы препроцессора**

Директивы препроцессора — это команды, которые обрабатываются до компиляции кода. Они начинаются с символа # и выполняют различные функции:

* #include: используется для включения содержимого одного файла в другой. Например:

#include <iostream> // Включает стандартную библиотеку ввода-вывода

#include "myheader.h" // Включает пользовательский заголовочный файл

* #define: используется для определения макросов, которые могут быть заменены на заданные значения или выражения. Например:

#define PI 3.14159

* #pragma once : используются для предотвращения множественного включения одного и того же заголовочного файла. Это называется "защита от повторного включения". Пример:

#pragma once

*// Содержимое заголовочного файла*

**Постановка задачи.**

Необходимо написать программу, которая содержит в себе инфраструктуру переключения между практическими работами (интерактивное меню). В меню должна быть реализована возможность многократно запускать разные лабораторные работы.

**Выполнение работы.**

Код программы представлен в приложении А.

| Ввод пользователем и обработка данных | Работа алгоритма и вывод на экран |
| --- | --- |
| Меню | |
| При запуске программы пользователь видит меню. | Программа предоставляет пользователю выбор между запуском конкретной практической работы или выходом из программы: |
| Запуск практической работы | |
| Пользователь вводит номер практической работы для запуска. | Программа запрашивает ввод пользователя.  После ввода запускается соответствующая практическая работа:    После завершения выполненя практической работы интерфейс программы вновь предлагает пользовалю выбрать один из пунктов меню. |
| Выход из программы | |
| Пользователь вводит цифру «0» для завершения работы програмы. | Программа запрашивает ввод пользователя: |

**Выводы.**

В ходе выполнения курсовой работы была изучена структура многофайловой программы на C++, что позволило понять принципы организации кода и использования заголовочных файлов для объявления функций из разных модулей. Также было реализовано интерактивное меню, позволяющее пользователю выбирать различные функции программы.

Приложение А

рабочий код

**Файл main.cpp**

#include <iostream>

#include "1lab.h"

#include "2lab.h"

#include "3lab.h"

#include "4lab.h"

#include <clocale>

using namespace std;

int main() {

setlocale(LC\_ALL, "RUS");

int num=1;

cout << "Введите номер работы или 0 для завершения: ";

while (num != 0) {

cin >> num;

switch (num) {

case 1:

run\_lab1();

break;

case 2:

run\_lab2();

break;

case 3:

run\_lab3();

break;

case 4:

run\_lab4();

break;

default:

cout << "Неверный номер работы";

break;

}

system("cls");

cout << "Введите номер работы или 0 для завершения: ";

}

return 0;

}

**Файл 1.cpp**

#include <iostream>

#include <conio.h>

#include <string>

using namespace std;

int run\_lab1() {

setlocale(0, "");

char num\_zad;

cout << "\033[32m"<< "Введите номер задания или \"q\" для завершения работы: " << "\33[0m";

cin >> num\_zad;

if (num\_zad != '1' && num\_zad != '2' && num\_zad != '3' && num\_zad != '4' && num\_zad != 'q')

num\_zad = 'e';

while (num\_zad != 'q') {

// 1 задание

if (num\_zad=='1')

{

string h, n;

int t;

int mark=0;

cout << "\033[33m"<<"Введите тип данных: " << "\033[31m";

cin >> h;

if (h == "short" || h == "long") {

cin >> n;

t = 1;

}

else

t = 0;

if (t) {

if (n == "int" && h == "short") {

cout << "short int " << "\033[0m" << "- " << sizeof(short int) << " байт\n";

mark = 1;

}

if (n == "int" && h == "long") {

cout << "long int " << "\033[0m" << "- " << sizeof(long int) << " байт\n";

mark = 1;

}

if (n == "double" && h == "long") {

cout << "long double " << "\033[0m" << "- " << sizeof(long double) << " байт\n";

mark = 1;

}

}

else {

if (h == "int") {

cout << "int " << "\033[0m" << "- " << sizeof(int) << " байт\n";

mark = 1;

}

if (h == "float") {

cout << "float " << "\033[0m" << "- " << sizeof(float) << " байт\n";

mark = 1;

}

if (h == "double") {

cout << "double " << "\033[0m" << "- " << sizeof(double) << " байт\n";

mark = 1;

}

if (h == "char") {

cout << "char " << "\033[0m" << "- " << sizeof(char) << " байт\n";

mark = 1;

}

if (h == "bool") {

cout << "bool " << "\033[0m" << "- " << sizeof(bool) << " байт\n";

mark = 1;

}

if (mark==0)

cout << "\033[31mНекорректный ввод!\033[0m";

}

cout << endl;

cout << "\033[32m" << "Введите номер задания или \"q\" для завершения работы: " << "\33[0m";

cin >> num\_zad;

if (num\_zad != '1' && num\_zad != '2' && num\_zad != '3' && num\_zad != '4' && num\_zad != 'q')

num\_zad = 'e';

}

// 2 задание

if (num\_zad == '2')

{

int answer2;

int bit\_number2;

char bit\_value2;

int number1;

int mas2;

cout << "\033[33m" << "Введите целое число: "<<"\033[0m";

cin >> number1;

cout << endl;

unsigned int(mask1) = 1 << 31;

cout << "Двоичное представление в памяти: " << "\033[31m";

for (int i = 32; i != 0; --i) {

putchar(number1 & mask1 ? '1' : '0');

if (i == 32 || i % 8 == 1) {

cout << " ";

if (i == 32)

cout << "\033[35m";

}

mask1 >>= 1;

}

cout << endl << endl;

cout << "\033[0mЕсли хотите заменить бит, введите \033[33m1\033[0m, иначе введите \033[31m0\033[0m: ";

cin >> answer2;

while (answer2 == 1) {

cout << endl << "Введите номер бита: ";

cin >> bit\_number2;

cout << endl;

cout << "Введите на что заменить бит: ";

cin >> bit\_value2;

cout << endl;

mask1 = 1 << 31;

if ((bit\_value2 == '1') or (bit\_value2 == '0')) {

if (bit\_value2 == '1') {

mas2 = 1 << 32 - 1 - bit\_number2;

number1 = number1 ^ mas2;

cout << number1 << endl;

for (int i = 32; i != 0; --i) {

putchar(number1 & mask1 ? '1' : '0');

if (i == 32 || i % 8 == 1) {

cout << " ";

if (i == 32)

cout << "\033[35m";

}

mask1 >>= 1;

}

cout << "\033[31m";

}

else {

mas2 = ~(1 << 32 - 1- bit\_number2);

number1 = (number1 & mas2);

cout << number1 << endl;

for (int i = 32; i != 0; --i) {

putchar(number1 & mask1 ? '1' : '0');

if (i == 32 || i % 8 == 1) {

cout << " ";

if (i == 32)

cout << "\033[35m";

}

mask1 >>= 1;

}

cout << "\033[31m";

}

}

else {

cout << "\033[31mНекорректный ввод!\033[0m";

break;

}

cout << endl << endl << "Если хотите ещё раз заменить бит, введите 1, иначе введите \033[31m0\033[0m: ";

cin >> answer2;

}

cout << endl<<endl;

cout << "\033[32m" << "Введите номер задания или \"q\" для завершения работы: " << "\33[0m";

cin >> num\_zad;

if (num\_zad != '1' && num\_zad != '2' && num\_zad != '3' && num\_zad != '4' && num\_zad != 'q')

num\_zad = 'e';

}

// 3 задание

if (num\_zad == '3')

{

char mas3[33]{};

int answer3;

int bit\_number3;

char bit\_value3;

union {

int b;

float number2;

};

cout << "\033[33m" << "Введите вещественное число: "<<"\033[0m";

cin >> number2;

cout << endl;

unsigned int(mask2) = 1 << 31;

cout << "Двоичное представление в памяти: "<<"\033[31m";

for (int i = 32; i != 0; --i) {

putchar(b & mask2 ? '1' : '0');

if (b & mask2)

mas3[32 - i] = '1';

else

mas3[32- i] = '0';

if (i == 32 || i == 32 - 8 || i == 32 - 8 \* 2 || i == 32 - 8 \* 3) {

cout << " ";

if (i == 32)

cout << "\033[36m";

if (i==32-8)

cout << "\033[35m";

}

mask2 >>= 1;

}

cout << endl << endl;

cout << "\033[0mЕсли хотите заменить бит, введите \033[33m1\033[0m, иначе введите \033[31m0\033[0m: ";

cin >> answer3;

while (answer3 == 1) {

cout << endl << "Введите номер бита: ";

cin >> bit\_number3;

cout << endl;

cout << "Введите на что заменить бит: ";

cin >> bit\_value3;

cout << endl;

if ((bit\_value3 == '1') or (bit\_value3 == '0')) {

mas3[bit\_number3 - 1] = bit\_value3;

cout << "\033[31m";

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

cout << mas3[i];

if (i == 0 || i == 8 || i == 8 \* 2 || i == 8 \* 3) {

if (i == 0)

cout << "\033[36m";

if (i == 8)

cout << "\033[35m";

cout << " ";

}

}

}

else {

cout << "\033[31mНекорректный ввод!\033[0m";

break;

}

cout << endl <<endl<< "Если хотите ещё раз заменить бит, введите 1, иначе введите \033[31m0\033[0m: ";

cin >> answer3;

}

cout << endl << endl;

cout << "\033[32m" << "Введите номер задания или \"q\" для завершения работы: " << "\33[0m";

cin >> num\_zad;

if (num\_zad != '1' && num\_zad != '2' && num\_zad != '3' && num\_zad != '4' && num\_zad != 'q')

num\_zad = 'e';

}

//4 задание

if (num\_zad == '4') {

cout << endl;

char mas4[64];

int answer4;

int bit\_number4;

char bit\_value4;

union {

double c;

int d[2];

};

cout << "\033[33m" << "Введите вещественное число: " << "\033[0m";

cin >> c;

cout << endl<<"Двоичное представление в памяти: " << "\033[31m";

int db = 0;

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

unsigned int mask3 = 1 << 31;

for (int i = 32; i != 0; --i) {

db++;

putchar(d[j] & mask3 ? '1' : '0');

if (d[j] & mask3)

mas4[32 + 32 \* (1 - j) - i] = '1';

else

mas4[32 + 32 \* (1 - j) - i] = '0';

if (db == 1 || db == 12) {

cout << " ";

if (db==1)

cout << "\033[36m";

if (db == 12)

cout << "\033[35m";

}

mask3 >>= 1;

}

}

cout << endl<<endl;

cout << "\033[0mЕсли хотите заменить бит, введите \033[33m1\033[0m, иначе введите \033[31m0\033[0m: ";

cin >> answer4;

while (answer4 == 1) {

cout <<endl<< "Введите номер бита: ";

cin >> bit\_number4;

cout << endl;

cout << "Введите на что заменить бит: ";

cin >> bit\_value4;

cout << endl;

if ((bit\_value4 == '1') or (bit\_value4 == '0')) {

mas4[bit\_number4 - 1] = bit\_value4;

cout << "\033[31m";

for (int i = 0; i != 64; ++i) {

if (i == 1 || i == 12) {

cout << " ";

if (i == 1)

cout << "\033[36m";

if (i == 12)

cout << "\033[35m";

}

cout << mas4[i];

}

}

else {

cout << "\033[31mНекорректный ввод!\033[0m";

break;

}

cout << endl << endl<<"Если хотите ещё раз заменить бит, введите 1, иначе введите \033[31m0\033[0m: ";

cin >> answer4;

}

cout << endl << endl;

cout << "\033[32m" << "Введите номер задания или \"q\" для завершения работы: " << "\33[0m";

cin >> num\_zad;

if (num\_zad != '1' && num\_zad != '2' && num\_zad != '3' && num\_zad != '4' && num\_zad != 'q')

num\_zad = 'e';

}

if (num\_zad =='e') {

cout << "\033[31mНекорректный ввод!\033[0m";

cout << endl << endl;

cout << "\033[32m" << "Введите номер задания или \"q\" для завершения работы: " << "\33[0m";

cin >> num\_zad;

if (num\_zad != '1' && num\_zad != '2' && num\_zad != '3' && num\_zad != '4' && num\_zad != 'q')

num\_zad = 'e';

cin.sync();

}

}

return(0);

}**Файл 1lab.h**

#pragma once

int run\_lab1();

**Файл 2.cpp**

#include <iostream>

#include <ctime>

#include <chrono>

const int N = 100, mxel = 199, mnel = -99;

using namespace std;

using namespace chrono;

int arr[N], sort\_arr[N], sort\_arr1[N];

time\_point<steady\_clock, duration<\_\_int64, ratio<1, 1000000000>>> start\_time, end\_time;

nanoseconds result\_time;

void insertsort(int\* arr, int N) {

for (int i = 1; i < N; i++) {

if (arr[0] > arr[1]) swap(arr[0], arr[1]);

int mid = 0;

int lo = 0;

int hi = i;

while (lo < hi) {

mid = (lo + hi) / 2;

if (arr[i] > arr[mid]) lo = mid + 1;

if (arr[i] <= arr[mid]) hi = mid;

}

int j = i;

while (j > lo) {

swap(arr[j], arr[j - 1]);

j--;

}

}

}

void quicksort(int\* arr, int end, int begin)

{

int mid;

int f = begin;

int l = end;

mid = arr[(f + l) / 2];

while (f < l)

{

while (arr[f] < mid) f++;

while (arr[l] > mid) l--;

if (f <= l)

{

swap(arr[f], arr[l]);

f++;

l--;

}

}

if (begin < l) quicksort(arr, l, begin);

if (f < end) quicksort(arr, end, f);

}

int binarysearch(int\* arr, int value, int start, int end) {

if (end >= start) {

int mid = start + (end - start) / 2;

if (arr[mid] == value) {

return mid;

}

if (arr[mid] > value) {

return binarysearch(arr, value, start, mid - 1);

}

return binarysearch(arr, value, mid + 1, end);

}

return -1;

}

int binarysearch1(int\* arr, int value, int start, int end) {

if (end >= start) {

int mid = start + (end - start) / 2;

if (arr[mid] == value) {

return mid;

}

if (arr[mid] > value) {

return binarysearch(arr, value, start, mid - 1);

}

return binarysearch(arr, value, mid + 1, end);

}

return -10000;

}

void task1() {

cout << endl << "1 Задание: " << endl << endl;

srand(time(0));

cout << "Несортированный массив: ";

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

{

arr[i] = rand() % 19 -9;

cout << arr[i] << " ";

}

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

sort\_arr[i] = arr[i];

sort\_arr1[i] = arr[i];

}

cout << endl << endl;

}

void task11() {

cout << "Несортированный массив: ";

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

{

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

cout << arr[i] << " ";

}

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

sort\_arr[i] = arr[i];

sort\_arr1[i] = arr[i];

}

cout << endl << endl;

}

void task2() {

cout << endl << "2 Задание: " << endl << endl;

int end = N - 1, begin = 0;

start\_time = steady\_clock::now();

quicksort(sort\_arr, end, begin);

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << endl << "Отсортированный массив: ";

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

cout << sort\_arr[i] << " ";

cout << endl;

cout << "Время, затраченное на сортировку: " << result\_time.count() << " наносекунд" << endl;

cout << endl << endl;

}

void task21() {

int end = N - 1, begin = 0;

start\_time = steady\_clock::now();

quicksort(sort\_arr, end, begin);

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << endl << "Отсортированный массив: ";

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

cout << sort\_arr[i] << " ";

cout << endl;

cout << "Время, затраченное на сортировку: " << result\_time.count() << " наносекунд" << endl;

cout << endl << endl;

}

void task3() {

cout << endl << "3 Задание: " << endl << endl;

start\_time = steady\_clock::now();

int min = sort\_arr[0];

int max = sort\_arr[N - 1];

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

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

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

cout << "Время, затраченное на поиск в отсортированном массиве: " << result\_time.count() << " наносекунд" << endl;

cout << endl;

int mm = arr[0];

int mn = arr[0];

start\_time = steady\_clock::now();

for (int i = 1; i < N; i++) {

if (arr[i] > mn)

mn = arr[i];

if (arr[i] < mm)

mm = arr[i];

}

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << "Min: " << mm << "\n";

cout << "Max: " << mn << "\n";

cout << "Время, затраченное на поиск в неотсортированном массиве: " << result\_time.count() << " наносекунд" << endl << endl;

}

void task4() {

cout << endl << "4 Задание: " << endl << endl;

int min = sort\_arr[0];

int max = sort\_arr[N - 1];

int mid = (min + max) / 2;

int k = 0;

int d,d1;

cout <<"Среднее значение: " << mid<<endl<<endl;

cout << "Поиск индексов со средним значением в неотсортированном массиве: " ;

start\_time = steady\_clock::now();

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

if (arr[i] == mid) {

cout << i << " ";

k++;

}

}

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << endl;

cout << "Количество элементов со средним значением: " << k<< endl;

cout << "Время, затраченное на поиск в неотсортированном массиве: " << result\_time.count() << " наносекунд" << endl << endl;

k = 0;

cout << "Поиск индексов со средним значением в отсортированном массиве: ";

start\_time = steady\_clock::now();

d=binarysearch1(sort\_arr, mid, 0, N);

if (d > -10000) {

d1 = d - 1;

while ((sort\_arr[d] == mid)and(d<=N-1)) {

k++;

d++;

}

while ((sort\_arr[d1] ==mid)and(d1>=0)) {

k++;

d1--;

}

}

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << endl;

cout << "Количество элементов со средним значением: " << k << endl;

cout << "Время, затраченное на поиск в отсортированном массиве: " << result\_time.count() << " наносекунд" << endl << endl;

cout << endl;

}

void task5() {

cout << endl << "5 Задание: " << endl << endl <<"Введите число: ";

int a;

cin >> a;

int i = 0;

while (sort\_arr[i] < a) {

i++;

}

cout << endl << "Количество элементов, которые меньше числа " << a << " = " << i<<endl<<endl;

}

void task6() {

cout << endl << "6 Задание: " << endl << endl << "Введите число: ";

int b;

cin >> b;

int i = N-1;

while (sort\_arr[i] > b) {

i--;

}

cout << endl << "Количество элементов, которые больше числа " << b << " = " << N-1-i << endl << endl;

}

void task7() {

cout << endl << "7 Задание: " << endl << endl << "Введите число: ";

int c;

cin >> c;

cout << endl << "Обычный перебор: " << endl;

int d = -1;

start\_time = steady\_clock::now();

for (int i = 0; (sort\_arr[i] <= c) and (i<N); i++) {

if (sort\_arr[i] == c) {

d++;

break;

}

}

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

if (d == -1) {

cout << endl << "Введенного пользователем числа нет в отсортированном массиве" << endl;

}

else {

cout << endl << "Введенное пользователем число есть в отсортированном массиве" << endl;

}

cout << "Время, затраченное на поиск обычным перебором: " << result\_time.count() << " наносекунд" << endl << endl;

cout << endl << "Алгоритм бинарного поиска: " << endl;

d = -1;

start\_time = steady\_clock::now();

d = (binarysearch(sort\_arr, c, 0, N-1));

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

if (d == -1) {

cout << endl << "Введенного пользователем числа нет в отсортированном массиве" << endl;

}

else {

cout << endl << "Введенное пользователем число есть в отсортированном массиве" << endl;

}

cout << "Время, затраченное на бинарный поиск: " << result\_time.count() << " наносекунд" << endl << endl;

cout << endl;

}

void task8() {

cout << endl<<"8 Задание: " << endl << endl << "Введите индекс первого элемента: ";

int m1, m2;

cin >> m1;

cout << endl << "Введите индекс второго элемента: ";

cin >> m2;

start\_time = steady\_clock::now();

swap(arr[m1], arr[m2]);

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << "Измененный массив:" << endl;

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

{

cout << arr[i] << " ";

}

cout <<endl<< "Время, затраченное на обмен: " << result\_time.count() << " наносекунд" << endl << endl;

}

void task9() {

cout << endl << "9 Задание: " << endl << endl;

task11();

start\_time = steady\_clock::now();

insertsort(sort\_arr1, N);

end\_time = steady\_clock::now();

result\_time = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << "Сортировка вставками с использованием бинарного поиска: " << endl;

cout << endl << "Отсортированный массив: ";

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

cout << sort\_arr1[i] << " ";

cout << endl;

cout << "Время, затраченное на сортировку: " << result\_time.count() << " наносекунд" << endl;

cout << endl << endl;

cout << "Быстрая сортировка: " << endl;

task21();

}

int run\_lab2()

{

srand(time(nullptr));

task1();

int choice;

cout<< "1 - Создать массив;" << endl

<< "2 - Отсортировать массив;" << endl

<< "3 - Найти максимальный и минимальный элементы;" << endl

<< "4 - Выводит среднее значение, индексы равных ему элементов и количество;" << endl

<< "5 - Выводит количество элементов, которые меньше введенного числа;" << endl

<< "6 - Выводит количество элементов, которые больше введенного числа;" << endl

<< "7 - Поиск введенного числа;" << endl

<< "8 - Меняет местами элементы" << endl

<< "9 - Сортирует простыми вставками, используя алгоритм бинарного поиска" << endl;

do {

cout << "Выберите задание для выполнения (1-9) или 0 для завершения: " << endl<<endl;

cin >> choice;

switch (choice) {

case 1:

task1();

break;

case 2:

task2();

break;

case 3:

task3();

break;

case 4:

task4();

break;

case 5:

task5();

break;

case 6:

task6();

break;

case 7:

task7();

break;

case 8:

task8();

break;

case 9:

task9();

break;

case 0:

cout << "Завершение работы программы." << endl;

break;

default:

cout << "Некорректный ввод. Введите число от 0 до 9" << endl;

break;

}

} while (choice != 0);

return(0);

}

**Файл 2lab.h**

#pragma once

int run\_lab2();

**Файл 3lab.cpp**

#include <iostream>

#include <ctime>

#include <Windows.h>

using namespace std;

const int N = 6;

const int WIDTH = 3;

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

COORD destCoord;

void genmx(int matrix[][N], int x, int y, int offsetx, int offsety) {

\*(\*(matrix + y) + x) = (rand() % (N \* N) + 1);

destCoord.X = (x + offsetx) \* WIDTH;

destCoord.Y = y + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + y) + x);

Sleep(10);

}

void outmx(int matrix[][N], int offset\_x, int offset\_y) {

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

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

destCoord.X = (i + offset\_x) \* WIDTH;

destCoord.Y = j + offset\_y;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + i) + j);

}

}

}

void outmx1(float matrix[][N], int offset\_x, int offset\_y) {

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

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

destCoord.X = (i + offset\_x) \* WIDTH;

destCoord.Y = j + offset\_y;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + i) + j);

}

}

}

void ItF(const int intMatrix[][N], float floatMatrix[][N]) {

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

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

floatMatrix[i][j] = static\_cast<float>(intMatrix[i][j]);

}

}

}

void quicksort(int arr[][N], int end, int begin)

{

int mid;

int f = begin;

int l = end;

int ind = (f + l) / 2;

mid = (\*(\*(arr+ind%N)+ind/N));

while (f < l)

{

while (\*(\*(arr + f % N) + f / N) < mid) f++;

while (\*(\*(arr + l % N) + l / N) > mid) l--;

if (f <= l) {

swap(\*(\*(arr + f % N) + f / N), \*(\*(arr + l % N) + l / N));

f++;

l--;

}

}

if (begin < l) quicksort(arr, l, begin);

if (f < end) quicksort(arr, end, f);

}

void shake\_a(int matrix[][N], int offsetx, int offsety) {

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

for (int j = 0; j < N / 2; j++) {

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i + N / 2);

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i + N / 2);

}

}

}

void shake\_b(int matrix[][N], int offsetx, int offsety) {

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

for (int j = 0; j < N / 2; j++) {

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i + N / 2);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i);

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i + N / 2);

}

}

}

void shake\_c(int matrix[][N], int offsetx, int offsety) {

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

for (int j = 0; j < N / 2; j++) {

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i + N / 2);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i + N / 2);

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i);

}

}

}

void shake\_d(int matrix[][N], int offsetx, int offsety) {

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

for (int j = 0; j < N / 2; j++) {

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i + N / 2);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j) + i);

destCoord.X = (i + N / 2 + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i);

destCoord.X = (i + offsetx) \* WIDTH;

destCoord.Y = j + N / 2 + offsety;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*(\*(matrix + j + N / 2) + i + N / 2);

}

}

}

int run\_lab3() {

setlocale(LC\_ALL, "RUS");

cout << "Исходные матрицы:" << endl << endl;

int matrix[N][N], matrix1[N][N];

float matrix4[N][N], matrix5[N][N];

for (int i = 0; i <= N \* 2 - 1; i++) {

int a = i % 4;

switch (a) {

case(0):

for (int x = i / 4; x < N - i / 4; x++) {

genmx(matrix, x, i / 4, 0, 1);

}

break;

case(1):

for (int y = i / 4 + 1; y < N - i / 4; y++) {

genmx(matrix, N - i / 4 - 1, y, 0, 1);

}

break;

case(2):

for (int x = N-i / 4 - 2; x> 0+i / 4; x--) {

genmx(matrix, x, N-i/4-1, 0, 1);

}

break;

case(3):

for (int y = N - i / 4 - 1; y >0+ i/ 4; y--) {

genmx(matrix, i / 4, y, 0, 1);

}

break;

}

}

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

int a = i % 2;

switch (a) {

case(0):

for (int y = 0; y < N; y++) {

genmx(matrix1, i, y, N+1, 1);

}

break;

case(1):

for (int y = N-1; y >=0; y--) {

genmx(matrix1, i, y, N+1, 1);

}

break;

}

}

destCoord.X = 0;

destCoord.Y = N + 1;

SetConsoleCursorPosition(hStdout, destCoord);

cout << "2 задание"<<endl<<"Перестановки первой матрицы : ";

shake\_a(matrix, 0, N + 3);

shake\_b(matrix, N+1, N + 3);

shake\_c(matrix, (N+1)\*2, N + 3);

shake\_d(matrix, (N+1)\*3, N + 3);

destCoord.X = 0;

destCoord.Y = (N + 2)\*2;

SetConsoleCursorPosition(hStdout, destCoord);

cout << "Перестановки второй матрицы: ";

shake\_a(matrix1, 0, (N + 3)\*2);

shake\_b(matrix1, N + 1, (N + 3) \* 2);

shake\_c(matrix1, (N + 1) \* 2, (N + 3) \* 2);

shake\_d(matrix1, (N + 1) \* 3, (N + 3) \* 2);

destCoord.X = 0;

destCoord.Y = (N + 2) \* 3+1;

SetConsoleCursorPosition(hStdout, destCoord);

cout << "3 задание"<<endl<< "Отсортированные матрицы: ";

quicksort(matrix, N\*N-1,0);

quicksort(matrix1,N\*N-1,0);

outmx(matrix, 0, (N + 3) \* 3);

outmx(matrix1, N + 1, (N + 3) \* 3);

destCoord.X = 0;

destCoord.Y = (N + 2) \* 4 + 2;

SetConsoleCursorPosition(hStdout, destCoord);

system("pause");

system("cls");

cout <<"4 задание"<<endl<< "Манипуляция с элементами матрицы:"<<endl << "Исходные матрицы: ";

ItF(matrix, matrix4);

ItF(matrix1, matrix5);

outmx(matrix, 0, 3);

outmx(matrix1, N + 1, 3);

cout <<endl<< "Введите одну из операций(+, -, \*, /) : ";

char operation;

cin >> operation;

cout << endl << "Введите число: ";

int num;

cin >> num;

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

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

switch (operation) {

case '+':

\*(\*(matrix + i) + j) += num;

\*(\*(matrix1 + i) + j) += num;

break;

case '-':

\*(\*(matrix + i) + j) -= num;

\*(\*(matrix1 + i) + j) -= num;

break;

case '\*':

\*(\*(matrix + i) + j) \*= num;

\*(\*(matrix1 + i) + j) \*= num;

break;

case '/':

if (num == 0) {

\*(\*(matrix4 + i) + j) /= num;

\*(\*(matrix5 + i) + j) /= num;

}

else {

\*(\*(matrix + i) + j) /= num;

\*(\*(matrix1 + i) + j) /= num;

}

break;

}

}

}

destCoord.X = 0;

destCoord.Y = (N + 2) +4;

SetConsoleCursorPosition(hStdout, destCoord);

cout <<endl<< "Результат: ";

if ((operation == '/') && num == 0) {

outmx1(matrix4, 0, (N + 2) + 6);

outmx1(matrix5, N + 1, (N + 2) + 6);

}

else {

outmx(matrix, 0, (N + 2) + 6);

outmx(matrix1, N + 1, (N + 2) + 6);

}

cout << endl;

system("pause");

system("cls");

cout << "ИДЗ №10" << endl << "Матрица 3x3 со случайными числами от -30 до 30: " << endl;

int matrix2[3][3];

for (int i = 0; i < 3; ++i) {

for (int j = 0; j < 3; ++j) {

\*(\*(matrix2 + i) + j) = rand() % 61 - 30;

cout << \*(\*(matrix2 + i) + j) << '/t';

}

cout << endl;

}

int a = \*(\*(matrix2 + 0) + 0) \* \*(\*(matrix2 + 1) + 1) \* \*(\*(matrix2 + 2) + 2);

int b = \*(\*(matrix2 + 0) + 1) \* \*(\*(matrix2 + 1) + 2) \* \*(\*(matrix2 + 2) + 0);

int c = \*(\*(matrix2 + 0) + 2) \* \*(\*(matrix2 + 1) + 0) \* \*(\*(matrix2 + 2) + 1);

int d = \*(\*(matrix2 + 0) + 2) \* \*(\*(matrix2 + 1) + 1) \* \*(\*(matrix2 + 2) + 0);

int e = \*(\*(matrix2 + 0) + 0) \* \*(\*(matrix2 + 1) + 2) \* \*(\*(matrix2 + 2) + 1);

int f = \*(\*(matrix2 + 0) + 1) \* \*(\*(matrix2 + 1) + 0) \* \*(\*(matrix2 + 2) + 2);

cout << "Формула для вычисления определителя: (a11\*a22\*a33)+(a12\*a23\*a31)+(a13\*a21\*a32)-((a11\*a23\*a32)+(a12\*a21\*a33)+(a13\*a22\*a31))"<<endl;

Sleep(30);

cout << "1 промежуточный результат: (" << a << ")+(" << b << ")+(" << c << ")-((" << d << ")+(" << e << ")+(" << f << "))"<<endl;

cout << "2 промежуточный результат: " << a + b + c << "-" << d + e + f << endl;

Sleep(30);

cout << "Определитель = " << a + b + c - d - e - f<<endl;

system("pause");

return(0);

}

**Файл 3lab.h**

#pragma once

int run\_lab3();

**Файл lab4.cpp**

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <limits>

#include <windows.h>

using namespace std;

bool isrulower(char c) {

return (c >= 'а') && (c <= 'я');

}

bool isruupper(char c) {

return (c >= 'А') && (c <= 'Я');

}

int strsize(char a[]) {

int i = 0;

while (\*(a+i) != '\0') {

i++;

}

i++;

return(i);

}

char plusch(char a[], char b[]) {

int al = strsize(a) + strsize(b);

char c[10\*10];

int i = 0;

while (a[i] != '\0') {

c[i] = a[i];

i++;

}

int j = 0;

while (b[j] != '\0') {

c[i] = b[j];

i++;

j++;

}

c[i] = '\0';

cout << c;

return('0');

}

vector<int> prfx(string pat) {

int n = pat.length();

vector<int> lps(n, 0);

int i = 1;

int len = 0;

while (i < n) {

if (pat[i] == pat[len]) {

len++;

lps[i] = len;

i++;

}

else {

if (len != 0) {

len = lps[len - 1];

}

else {

lps[i] = 0;

i++;

}

}

}

return(lps);

}

vector<int> kmp(string str, string pstr) {

vector <int> occur;

vector<int> lps(prfx(pstr));

int n = str.length();

int m = pstr.length();

int i = 0;

int j = 0;

while (i < n) {

if (str[i] == pstr[j]) {

i++;

j++;

}

if (j == m) {

occur.push\_back(i - j);

j = lps[j - 1];

}

else if (i < n && str[i] != pstr[j]) {

if (j != 0) {

j = lps[j - 1];

}

else {

i++;

}

}

}

return (occur);

}

int linsearch(string str, string pstr) {

int n = str.length();

int m = pstr.length();

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

int j;

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

if (str[i + j] != pstr[j])

break;

}

if (j == m)

return i;

}

return -1;

}

string task2(string S) {

vector<int> spaces = kmp(S, " ");

int k = 0;

for (int i : spaces) {

S.erase(i - k, 1);

k++;

}

vector<int> dots = kmp(S, "..");

vector<int> rdots = kmp(S, "...");

k = 0;

for (int i : dots) {

for (int j : rdots) {

if (i == j) {

dots.erase(find(dots.begin(), dots.end(), i));

dots.erase(find(dots.begin(), dots.end(), i + 1));

k++;

}

}

}

k = 0;

for (int i : dots) {

S.erase(i - k, 1);

k++;

}

vector<int> comma = kmp(S, ",,");

k = 0;

for (int i : comma) {

S.erase(i - k, 1);

k++;

}

vector<int> excl = kmp(S, "!!");

k = 0;

for (int i : excl) {

S.erase(i - k, 1);

k++;

}

vector<int> quest = kmp(S, "??");

k = 0;

for (int i : quest) {

S.erase(i - k, 1);

k++;

}

vector<int> dash = kmp(S, "--");

k = 0;

for (int i : dash) {

S.erase(i - k, 1);

k++;

}

vector<int> colon = kmp(S, "::");

k = 0;

for (int i : colon) {

S.erase(i - k, 1);

k++;

}

vector<int> semicolon = kmp(S, ";;");

k = 0;

for (int i : semicolon) {

S.erase(i - k, 1);

k++;

}

/\* for (int i = 0; i < S.length(); i++) {

if ((isruupper(S[i]) && isruupper(S[i + 1])) || (isrulower(S[i]) && isruupper(S[i + 1]))) {

S[i + 1] = tolower(S[i + 1]);

}

}\*/

for (int i = 0; i < S.length(); i++) {

if ((isupper(S[i]) && isupper(S[i + 1])) || (islower(S[i]) && isupper(S[i + 1]))) {

S[i + 1] = tolower(S[i + 1]);

}

}

return(S);

}

void task3(string S) {

int m[3] = { 0,0,0 };

int kr = 0;

int figure = 0;

int letter = 0;

for (int i = 0; i < S.length(); i++) {

if ((S[i] == ' ') || (S[i] == '.') || (S[i] == ',') || (S[i] == '?') || (S[i] == '!') || (S[i] == '-') || (S[i] == ':') || (S[i] == ';')) {

if (figure && letter) {

kr = 1;

m[2] += 1;

}

else if (figure) {

m[1] += 1;

}

else if (letter) {

m[0] += 1;

}

letter = 0;

figure = 0;

}

else if ('0' <= S[i] && S[i] <= '9') {

figure = 1;

}

else {

letter = 1;

}

}

cout << "Количество слов, состоящих только из букв: " << m[0] << endl;

cout << "Количество слов, состоящих только из цифр: " << m[1] << endl;

cout << "Количество слов, состоящих как из букв, так и из цифр: " << m[2] << endl;

}

string task4(string S) {

char n;

char l;

for (int i = 0; i < S.length(); i++) {

n = S[i];

l = n;

switch (n) {

case('0'):

l = 'a';

break;

case('1'):

l = 'b';

break;

case('2'):

l = 'c';

break;

case('3'):

l = 'd';

break;

case('4'):

l = 'e';

break;

case('5'):

l = 'f';

break;

case('6'):

l = 'g';

break;

case('7'):

l = 'h';

break;

case('8'):

l = 'i';

break;

case('9'):

l = 'j';

break;

}

S[i] = l;

}

return(S);

}

void task5(string S) {

cout << endl << "Задание 5:" << endl;

char n = '2';

string s;

int result;

vector<int>occur;

while (n != '0') {

do {

cout << "Если хотите найти подстроку, введите 1, иначе 0." << endl;

cin >> n;

} while (n != '1' && n != '0');

cin.ignore((numeric\_limits<streamsize>::max)(), '\n');

switch (n) {

case('1'):

cout << "Введите подстроку, которую хотите найти." << endl;

getline(cin, s);

cout << endl << "Алгоритм линейного поиска:" << endl;

result = linsearch(S, s);

if (result == -1)

cout << "Подстроки нет в строке." << endl;

else

cout << "Подстрока найдена в строке на позиции: " << result << endl;

cout << endl << "Алгоритм Кнута-Мориса-Пратта:" << endl;

occur = kmp(S, s);

if (occur.empty()) {

cout << "Подстроки нет в строке." << endl << endl;

}

else {

cout << "Подстрока найдена в строке на позициях: ";

for (int i : occur) {

cout << i << " ";

}

cout << endl << endl;

}

break;

case('0'):

break;

}

}

}

int run\_lab4() {

int l;

char a[] = { "fox jumps over the lazy dog a"};

char b[] = { "the lazy dog " };

plusch(a, b);

l=strsize(a);

cout << l<<endl;

string S2, S3;

cout << "Выберите способ ввода текста:" << endl;

cout << "1. С клавиатуры" << endl;

cout << "2. Из файла" << endl;

int ans = -1;

do {

cout << "Ваш выбор: ";

cin >> ans;

} while (ans != 1 && ans != 2);

cin.ignore((numeric\_limits<streamsize>::max)(), '\n');

string S;

if (ans == 1) {

cout << "Введите текст" << endl;

getline(cin, S);

}

else {

cout << "Введите путь до файла с текстом: ";

string path;

getline(cin, path);

fstream file(path, ios\_base::in);

if (!file.is\_open()) {

cout << "Не удалось открыть файл, завершаю выполнение.";

return 1;

}

getline(file, S);

file.close();

}

cout << endl << "Задание 3:" << endl;

task3(S);

cout << endl << "Задание 2:" << endl;

S2 = task2(S);

cout << S2 << endl;

cout << endl << "Задание 4:" << endl;

S3 = task4(S2);

cout << S3 << endl;

task5(S3);

}

**Файл 4lab.h**

#pragma once

int run\_lab4();