**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по Курсовой работе**

**по дисциплине «Программирование»**

Тема: " [ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ С++](https://stepik.org/course/82772)"

|  |  |  |
| --- | --- | --- |
| Студент гр. 0324 |  | Гордиенко Т.Е. |
| Преподаватель |  | Глущенко А.Г |

Санкт-Петербург

2020

#include <iostream>

#include <chrono>

#include <random>

#include <Windows.h>

#include <fstream>

#include <string>

using namespace std;

//WORK 1

void memory()

{

std::cout << "Объём памяти под int = " << sizeof(int) << " байта" << '\n';

std::cout << "Объём памяти под short int = " << sizeof(short int) << " байта" << '\n';

std::cout << "Объём памяти под long int = " << sizeof(long int) << " байта" << '\n';

std::cout << "Объём памяти под float = " << sizeof(float) << " байта" << '\n';

std::cout << "Объём памяти под double = " << sizeof(double) << " байт" << '\n';

std::cout << "Объём памяти под long double = " << sizeof(long double) << " байт" << '\n';

std::cout << "Объём памяти под char = " << sizeof(char) << " байт" << '\n';

std::cout << "Объём памяти под bool = " << sizeof(bool) << " байт" << '\n' << '\n';

}

void code\_int()

{

int num, mask;

mask = INT\_MIN;

std::cout << "Число - ";

std::cin >> num;

putchar(num & mask ? '1' : '0');

std::cout << " ";

for (int i = 1; i < sizeof(int) \* 8; i++) {

putchar(num & mask ? '1' : '0');

num <<= 1;

}

std::cout << " - код введённого числа" << '\n' << '\n';

}

void code\_float()

{

union {

long int tool;

float num;

};

int mask = INT\_MIN;

std::cout << "Число - ";

std::cin >> num;

putchar(tool & mask ? '1' : '0');

tool <<= 1;

std::cout << " ";

for (int i = 1; i < 9; i++) {

putchar(tool & mask ? '1' : '0');

tool <<= 1;

}

std::cout << " ";

for (int i = 9; i < 32; i++) {

putchar(tool & mask ? '1' : '0');

tool <<= 1;

}

std::cout << " - его код" << '\n' << '\n';

}

void code\_double()

{

union {

long int tool[2];

double num;

};

std::cout << "Число - ";

std::cin >> num;

int mask = INT\_MIN, a;

a = mask & tool[1];

if (a == mask) {

std::cout << "1 ";

}

else {

std::cout << "0 ";

}

tool[1] <<= 1;

for (int i = 1; i != 12; i++) {

a = tool[1] & mask;

if (a == mask) {

std::cout << '1';

}

else {

std::cout << '0';

}

tool[1] <<= 1;

}

std::cout << " ";

for (int i = 12; i != 32; i++) {

a = tool[1] & mask;

if (a == mask) {

std::cout << '1';

}

else {

std::cout << '0';

}

tool[1] <<= 1;

}

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

a = tool[0] & mask;

if (a == mask) {

std::cout << '1';

}

else {

std::cout << '0';

}

tool[0] <<= 1;

}

std::cout << " - его код" << '\n' << '\n';

}

int work1() {

setlocale(0, "");

int nomer;

start\_work1:

std::cout << "Выберите номер раздела" << '\n' << '\n';

std::cout << "1) Объём памяти под разные типы данных " << '\n';

std::cout << "2) Код числа int" << '\n';

std::cout << "3) Код числа float" << '\n';

std::cout << "4) Код числа double" << '\n' << '\n';

std::cin >> nomer;

std::cout << '\n';

switch (nomer) {

case 1:

memory();

break;

case 2:

code\_int();

break;

case 3:

code\_float();

break;

case 4:

code\_double();

break;

default:

std::cout << "Это уже не к нам)";

break;

}

std::cout << '\n' << "Если нужно что-то ещё - введите '1', если нет - что угодно, но не '1') " << '\n' << '\n';

int replay;

std::cin >> replay;

switch (replay) {

case 1:

goto start\_work1;

break;

default:

return 0;

break;

}

}

//WORK 2

void create\_array(int array[], int SIZE) {//Присваивание рандомных значений

long seed = chrono::system\_clock::now().time\_since\_epoch().count();

default\_random\_engine rand(seed);

uniform\_int\_distribution<int> d(-99, 99);

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

array[i] = d(rand);

}

}

void bubble\_sort(int array[], int SIZE) {//Сортировка пузырьклм

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

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

for (int j = 0; j != i - 1; j++) {

if (array[j] > array[j + 1]) {

swap(array[j], array[j + 1]);

}

}

}

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

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

cout << time.count() << " секунд на сортировку пузырьком" << '\n';

}

void shaker\_sort(int array[], int SIZE) {//сортировка шейкером

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

for (int i = 0; i != SIZE / 2; i++) {

for (int j = 0 + i; j != SIZE - 1 - i; j++) {

if (array[j] > array[j + 1]) {

swap(array[j], array[j + 1]);

}

}

for (int j = SIZE - 2 - i; j != 0 + i; j--) {

if (array[j] < array[j - 1]) {

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

}

}

}

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

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

cout << time.count() << " секунд на сортировку шэйкером" << '\n';

}

void comb\_sort(int array[], int SIZE) {//Сортировка расчёской

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

for (int i = SIZE - 1; i != 0; i--) {

for (int j = 0; j != SIZE - i; j++) {

if (array[j] > array[j + i]) {

swap(array[j], array[j + i]);

}

}

}

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

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

cout << time.count() << " секунд на сортировку расчёской" << '\n';

}

void insert\_sort(int array[], int SIZE) {//Сортировка вставками

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

int j;

for (int i = 1; i != SIZE; i++) {

if (array[i] < array[i - 1]) {

j = i;

while (array[j] < array[j - 1] && j != 0) {

std::swap(array[j], array[j - 1]);

j--;

}

}

}

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

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

cout << time.count() << " секунд на сортировку вставками" << '\n';

}

void quick\_sort(int array[], int first, int last) {//Бинарная сортировка

if (first < last) {

int opora = array[first], swap\_index = first;

for (int i = first + 1; i != last + 1; i++) {

if (array[i] <= opora) {

swap\_index++;

swap(array[i], array[swap\_index]);

}

}

swap(array[first], array[swap\_index]);

quick\_sort(array, first, swap\_index - 1);

quick\_sort(array, swap\_index + 1, last);

}

}

void min\_max(int array[], int SIZE) {//минимум и максимум

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

int min = 100, max = -100;

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

if (array[i] > max) {

max = array[i];

}

if (array[i] < min) {

min = array[i];

}

}

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

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

cout << time.count() << " секунд на выполнение" << '\n';

cout << "min = " << min << " max = " << max << '\n';

}

void mid(int array[], int SIZE) {//среднее минимума и максимума

int min = 100, max = -100, mid, quantity = 0;

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

if (array[i] > max) {

max = array[i];

}

if (array[i] < min) {

min = array[i];

}

}

mid = (min + max) / 2;

cout << "Среднее значение = " << mid << '\n';

cout << "Индексы элементов: ";

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

if (array[i] == mid) {

cout << " " << i;

quantity++;

}

}

cout << '\n' << "Всего - " << quantity << '\n';

}

void les(int array[], int SIZE) {//меньше А

comb\_sort(array, SIZE);

int a;

cout << "А = ";

cin >> a;

if (array[SIZE - 1] >= a) {

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

if (array[i] >= a) {

cout << i << " чисел меньше А" << '\n';

break;

}

}

}

else {

cout << SIZE << " чисел меньше А" << '\n';

}

}

void more(int array[], int SIZE) {//больше В

comb\_sort(array, SIZE);

int b;

cout << "B = ";

cin >> b;

if (array[0] > b) {

cout << SIZE << " чисел больше В";

}

else if (array[SIZE - 1] <= b) {

cout << "таких чисел нет";

}

else {

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

if (array[i] > b) {

cout << SIZE - i << " чисел больше В";

break;

}

}

}

}

void z\_in\_array(int array[], int SIZE) {//z в массиве

comb\_sort(array, SIZE);

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

int z;

cout << "Z = ";

cin >> z;

bool z\_in = 0;

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

if (array[i] == z) {

z\_in = 1;

break;

}

}

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

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

cout << time.count() << " секунд на перебор" << '\n';

if (z\_in) {

cout << "true" << '\n';

}

else {

cout << "false" << '\n';

}

z\_in = 0;

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

int i = SIZE / 2, quantity = SIZE, log2 = 1, n2 = 2, temp;

while (n2 <= SIZE) {

n2 \*= 2;

log2++;

}

for (int j = 0; j != log2; j++) {

if (array[i] == z) {

z\_in = 1;

break;

}

else if (array[i] < z) {

i = (quantity + i) / 2;

}

else if (array[i] > z) {

temp = i;

i = i - (quantity - i) / 2;

quantity = temp;

}

}

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

time = end - start;

cout << time.count() << " секунд на бинарный поиск" << '\n';

if (z\_in) {

cout << "true" << '\n';

}

else {

cout << "false" << '\n';

}

}

void swaping(int array[], int SIZE) {//Поменять n-й и m-й местами

int n, m;

Try\_again\_work2:

cout << "n = ";

cin >> n;

cout << "m = ";

cin >> m;

if (n < 0 || n > SIZE - 1 || m < 0 || m > SIZE - 1) {

cout << "try again" << '\n' << '\n';

goto Try\_again\_work2;

}

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

swap(array[n], array[m]);

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

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

cout << time.count() << " секунд на перестановку";

}

void umensh\_na\_k(int array[], int SIZE) {//уменьшить на k

int k;

cout << "k = ";

cin >> k;

for (int i = 1; i <= SIZE; i += 2) {

array[i] -= k;

}

}

void umnoj\_na\_random(int array[], int SIZE) {//умножить на рандом

long seed = chrono::system\_clock::now().time\_since\_epoch().count();

default\_random\_engine rand(seed);

uniform\_int\_distribution<int> d(1, 9);

for (int i = 1; i <= SIZE; i += 2) {

array[i] \*= d(rand);

}

}

void delitsya(int array[], int SIZE) {//сколько делится на k

int k;

cout << "K = ";

cin >> k;

int kol = 0;

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

if (array[i] % k == 0) {

kol++;

}

}

cout << kol << " чисел делится на " << k;

}

int work2() {

setlocale(0, "");//устанавливаем язык по умолчанию

int array[100], punkt;

chrono::system\_clock::time\_point start;

chrono::system\_clock::time\_point end;

chrono::duration<double> time;

cout << "0) Выход" << '\n';

cout << "1) Создать массив из рандомных чисел" << '\n';

cout << "2) Отсортировать массив" << '\n';//Время?

cout << "3) Найти max и min" << '\n';//Время? Сорт vs Не сорт

cout << "4) Среднее значение max и min, индексы равных ему чисел и их количество" << '\n';

cout << "5) Количество чисел, меньше A" << '\n';//В отсорт

cout << "6) Количесво чисел, больше B" << '\n';//В отсорт

cout << "7) Есть ли число Z в массиве" << '\n';//Время? Бинарный vs Перебор

cout << "8) Поменять местами n-й и m-й элемент" << '\n';//Время?

cout << "9) Вывести массив на экран" << '\n';

cout << "10) Нечётные уменьшить на k" << '\n';

cout << "11) Нечётные умножь на рандом" << '\n';

cout << "12) Сколько чисел делятся на k (k от 1 до 9)" << '\n' << '\n';

start\_work2:

cout << "Пункт - ";

cin >> punkt;

cout << '\n';

switch (punkt)

{

case 0:

return 0;

break;

case 1:

create\_array(array, sizeof(array) / 4);

break;

case 2:

Try\_again2\_work2:

int sort;

cout << "Выберити тип сортировки" << '\n' << '\n';

cout << "1) Сортировка пузырьком" << '\n';

cout << "2) Сортировка шэйкером" << '\n';

cout << "3) Сортировка расчёской" << '\n';

cout << "4) Сортировка вставками" << '\n';

cout << "5) Бинарная сортировкка" << '\n' << '\n';

cin >> sort;

switch (sort) {

case 1:

bubble\_sort(array, sizeof(array) / 4);

break;

case 2:

shaker\_sort(array, sizeof(array) / 4);

break;

case 3:

comb\_sort(array, sizeof(array) / 4);

break;

case 4:

insert\_sort(array, sizeof(array) / 4);

break;

case 5:

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

quick\_sort(array, 0, sizeof(array) / 4 - 1);

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

time = end - start;

cout << time.count() << " секунд на сортровку" << '\n';

break;

default:

cout << "try againt" << '\n' << '\n';

goto Try\_again2\_work2;

break;

}

break;

case 3:

min\_max(array, sizeof(array) / 4);

break;

case 4:

mid(array, sizeof(array) / 4);

break;

case 5:

les(array, sizeof(array) / 4);

break;

case 6:

more(array, sizeof(array) / 4);

break;

case 7:

z\_in\_array(array, sizeof(array) / 4);

break;

case 8:

swaping(array, sizeof(array) / 4);

break;

case 9:

for (int i = 0; i != sizeof(array) / 4; i++) {

cout << array[i] << " ";

}

cout << '\n' << '\n';

break;

case 10:

umensh\_na\_k(array, sizeof(array) / 4);

break;

case 11:

umnoj\_na\_random(array, sizeof(array) / 4);

break;

case 12:

delitsya(array, sizeof(array) / 4);

break;

default:

cout << "try again" << '\n' << '\n';

goto start\_work2;

break;

}

goto start\_work2;

}

//WORK 3

void gotoxy(int x, int y) {//перемещение курсора консоли на координату {x,y}

COORD p = { x, y };

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), p);

}

void spiral(int size, int \*num, int \*matrix) {//присваивание и вывод спиралькой

system("cls");

long seed = chrono::system\_clock::now().time\_since\_epoch().count();

default\_random\_engine rand(seed);

uniform\_int\_distribution<int> d(1, size \* size);

int s = 1, left = -1, up = -1, right, down, x, y, temp;

down = right = size;

x = 0;

y = 0;

gotoxy(x, y);

while (s <= size \* size) {

for (int i = left + 1; i < right; i++) {

temp = d(rand);

cout << temp;

\*num = temp;

num++;

s++;

x += 5;

gotoxy(x, y);

Sleep(100);

}

up++;

if (s > size\*size) {

break;

}

x -= 5;

num--;

y += 3;

num += 100;

gotoxy(x, y);

for (int i = up + 1; i < down; i++) {

cout << s;

\*num = s;

s++;

y += 3;

num += 100;

gotoxy(x, y);

Sleep(100);

}

right--;

if (s > size\*size) {

break;

}

y -= 3;

num -= 100;

x -= 5;

num--;

gotoxy(x, y);

for (int i = right - 1; i > left; i--) {

cout << s;

\*num = s;

s++;

x -= 5;

num--;

gotoxy(x, y);

Sleep(100);

}

down--;

if (s > size\*size) {

break;

}

x += 5;

num++;

y -= 3;

num -= 100;

gotoxy(x, y);

for (int i = down - 1; i > up; i--) {

cout << s;

\*num = s;

s++;

y -= 3;

num -= 100;

gotoxy(x, y);

Sleep(100);

}

left++;

y += 3;

num += 100;

x += 5;

num++;

gotoxy(x, y);

}

num = matrix;

x = 0;

y += ((size / 2) + 1) \* 3;

gotoxy(x, y);

}

void serpentine(int size, int \*num, int \*matrix) {//присваивание и вывод серпантином

system("cls");

int s = 1, x = 0, y = 0, temp;

long seed = chrono::system\_clock::now().time\_since\_epoch().count();

default\_random\_engine rand(seed);

uniform\_int\_distribution<int> d(1, size \* size);

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

if (i % 2 == 0) {

for (int j = 0; j != size; j++) {

temp = d(rand);

cout << temp;

\*num = temp;

s++;

y += 3;

num += 100;

gotoxy(x, y);

Sleep(100);

}

y -= 3;

num -= 100;

x += 5;

num++;

gotoxy(x, y);

}

else {

for (int j = 0; j != size; j++) {

cout << s;

\*num = s;

s++;

y -= 3;

num -= 100;

gotoxy(x, y);

Sleep(100);

}

y += 3;

num += 100;

x += 5;

num++;

gotoxy(x, y);

}

}

num = matrix;

if (size % 2 == 0) {

y += 3 \* size + 3;

x = 0;

}

else {

y += 3;

x = 0;

}

gotoxy(x, y);

}

void swap\_clock(int size, int \*num, int \*matrix) {//смена по часовой стрелке

int temp;

for (int i = 0; i != size / 2; i++) {

for (int j = 0; j != size / 2; j++) {

temp = \*num;

\*num = \*(num + (100 \* (size / 2)));

\*(num + (100 \* (size / 2))) = \*(num + (100 \* (size / 2)) + (size / 2));

\*(num + (100 \* (size / 2)) + (size / 2)) = \*(num + (size / 2));

\*(num + (size / 2)) = temp;

num++;

}

num -= size / 2;

num += 100;

}

num = matrix;

}

void swap\_x(int size, int \*num, int \*matrix) {//смена крест на крест

int temp;

for (int i = 0; i != size / 2; i++) {

for (int j = 0; j != size / 2; j++) {

temp = \*num;

\*num = \*(num + 100 \* (size / 2) + (size / 2));

\*(num + 100 \* (size / 2) + (size / 2)) = temp;

num++;

}

num -= size / 2;

num += 100;

}

for (int i = size / 2; i != size; i++) {

for (int j = 0; j != size / 2; j++) {

temp = \*num;

\*num = \*(num + (size / 2) - 100 \* (size / 2));

\*(num + (size / 2) - 100 \* (size / 2)) = temp;

num++;

}

num -= size / 2;

num += 100;

}

num = matrix;

}

void swap\_up\_down(int size, int \*num, int \*matrix) {//смена верх и низ

int temp;

for (int i = 0; i != size / 2; i++) {

for (int j = 0; j != size; j++) {

temp = \*num;

\*num = \*(num + 100 \* (size / 2));

\*(num + 100 \* (size / 2)) = temp;

num++;

}

num -= size;

num += 100;

}

num = matrix;

}

void swap\_left\_right(int size, int \*num, int \*matrix) {//смена лево и право

int temp;

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

for (int j = 0; j != size / 2; j++) {

temp = \*num;

\*num = \*(num + (size / 2));

\*(num + (size / 2)) = temp;

num++;

}

num -= size / 2;

num += 100;

}

num = matrix;

}

void sort(int size, int \*num, int \*matrix) {//сортировка

int temp, range = 100 - size + 1;

for (int j = size \* size; j != 0; j--) {

for (int i = 1; i != j; i++) {

if (i % size != 0) {

if (\*num > \*(num + 1)) {

temp = \*num;

\*num = \*(num + 1);

\*(num + 1) = temp;

}

num++;

}

else {

if (\*num > \*(num + range)) {

temp = \*num;

\*num = \*(num + range);

\*(num + range) = temp;

}

num += range;

}

}

num = matrix;

}

num = matrix;

}

void change(int size, int \*num, int \*matrix) {// + - / \*

char action;

int chislo;

try\_again2\_work3:

cout << "Знак действия ";

cin >> action;

cout << "Число ";

cin >> chislo;

switch (action) {

case '\*':

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

\*num \*= chislo;

num++;

}

break;

case '/':

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

\*num /= chislo;

num++;

}

break;

case '+':

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

\*num += chislo;

num++;

}

break;

case '-':

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

\*num -= chislo;

num++;

}

break;

default:

cout << "try again";

goto try\_again2\_work3;

}

num = matrix;

}

void print(int size, int \*num, int \*matrix) {//вывод матрицы

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

for (int j = 0; j != size; j++) {

if (\*num < 10) {

cout << \*num << " ";

}

else {

cout << \*num << " ";

}

num++;

}

num -= size;

num += 100;

cout << '\n';

}

num = matrix;

}

int work3() {

setlocale(0, "");

int size, action;

int matrix[100][100];

int \*num = \*matrix;

cout << "Введите размер массива (n) ";

cin >> size;

try\_again\_work3:

cout << "Выбеирте действие: " << '\n';

cout << "1) Вывести массив спиралькой и присвоить рандомные значения" << '\n';

cout << "2) Вывести массив серпантином и присвоить рандомные значения" << '\n';

cout << "3) Поменять блоки по часовой стрелке" << '\n';

cout << "4) Поменять блоки крест на крест" << '\n';

cout << "5) Поменять верхний и нижний блок" << '\n';

cout << "6) Поменять левый и правый блок" << '\n';

cout << "7) Отсортировать матрицу по возрастанию" << '\n';

cout << "8) Умножить/разделить/увеличить/уменьшить элементы на число А" << '\n';

cout << "9) Вывести матрицу" << '\n';

cout << "0) Выход" << '\n' << '\n';

cin >> action;

switch (action) {

case 1:

spiral(size, num, \*matrix);

break;

case 2:

serpentine(size, num, \*matrix);

break;

case 3:

swap\_clock(size, num, \*matrix);

break;

case 4:

swap\_x(size, num, \*matrix);

break;

case 5:

swap\_up\_down(size, num, \*matrix);

break;

case 6:

swap\_left\_right(size, num, \*matrix);

break;

case 7:

sort(size, num, \*matrix);

break;

case 8:

change(size, num, \*matrix);

break;

case 9:

print(size, num, \*matrix);

break;

case 0:

return 0;

break;

default:

cout << "try again" << '\n';

goto try\_again\_work3;

}

goto try\_again\_work3;

}

//WORK 4

void read\_file(char s[]) {

char stroka[1000];

string path = "Text.txt";

ifstream file;

file.open(path);

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

std::cout << "ошибка открытия файла";

return;

}

else {

file.getline(s, 1000, '\0');

}

}

int work4() {

setlocale(0, "");

char stroka[1000];

std::cout << "Способ чтения: " << '\n';

std::cout << "1) В консоли" << '\n';

std::cout << "2) Из файла" << '\n';

int var;

cin >> var;

switch (var) {

case 1:

cin.getline(stroka, 1000, '.');

break;

case 2:

read\_file(stroka);

break;

default:

std::cout << "error" << '\n';

return 1;

break;

}

string s = stroka;

//Редактирование строки

bool slovo = 0;

if ((s[0] >= 'A' && s[0] <= 'z') || (s[0] >= 'А' && s[0] <= 'я')) {

slovo = 1;

}

for (int i = 1; i != s.length() - 1; i++) {

if ((s[i] < 'A' || s[i] > 'Z') && (s[i] < 'a' || s[i] > 'z') && (s[i] < 'А' || s[i] > 'я')) {

slovo = 0;

if (s[i] == '.' && s[i - 1] == '.' && s[i + 1] == '.') {

i++;

}

else if (s[i] == s[i - 1]) {

s.erase(i, 1);

i--;

}

}

else {

if (s[i] >= 'A' && s[i] <= 'Z' && slovo == 1) {

s[i] += 'Z' - 'A' + 7;

}

else if (s[i] >= 'А' && s[i] <= 'Я' && slovo == 1) {

s[i] += 'Я' - 'А' + 1;

}

slovo = 1;

}

}

std::cout << s << '\n' << '\n';

//Вывод слов, в которых первая буква встречается ещё раз

int pos = 0;

char temp = '\0';

slovo = 0;

for (int i = 0; i <= s.length() - 1; i++) {

if (slovo == 0 && ((s[i] >= 'A' && s[i] <= 'Z') || (s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'А' && s[i] <= 'я'))) {

slovo = 1;

temp = s[i];

pos = i;

continue;

}

if (s[i] == temp && slovo == 1) {

while (((s[pos] >= 'A' && s[pos] <= 'Z') || (s[pos] >= 'a' && s[pos] <= 'z') || (s[pos] >= 'А' && s[pos] <= 'я')) && pos <= s.length() - 1) {

cout << s[pos];

pos++;

}

slovo = 0;

i = pos;

cout << '\n';

}

else if (s[i] == ' ') {

slovo = 0;

}

}

cout << '\n';

//Перемещение чисел из слов в их конец

slovo = 0;

int temp\_i;

string temp\_s;

for (int i = 0; i <= s.length() - 1; i++) {

if (((s[i] >= 'A' && s[i] <= 'Z') || (s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'А' && s[i] <= 'я'))) {

slovo = 1;

}

else if (slovo == 1 && s[i] >= '0' && s[i] <= '9') {

temp\_s.append(1, s[i]);

s.erase(i, 1);

i--;

}

else {

slovo = 0;

s.insert(i, temp\_s);

temp\_s.clear();

}

}

cout << s << '\n';

//Поиск подстрок

cout << "Введите подстроку" << '\n';

getline(cin, temp\_s);

getline(cin, temp\_s);

slovo = 1;

pos = 0;

for (int i = 0; i != s.length() - temp\_s.length(); i++) {

for (int j = i; j != i + temp\_s.length(); j++) {

if (s[j] == temp\_s[pos]) {

pos++;

}

else {

pos = 0;

slovo = 0;

break;

}

}

if (slovo == 1) {

cout << i << " ";

pos = 0;

}

else {

slovo = 1;

}

}

return 0;

}

int main() {

setlocale(NULL, "");

int n;

cout << "Выберите один из вариантов" << '\n';

cout << "1) ПР1 Типы данных и их представление" << '\n';

cout << "2) ПР2 Одномерные статические массивы" << '\n';

cout << "3) ПР3 Матрицы" << '\n';

cout << "4) ПР4 Строки" << '\n';

cout << "0) Выход" << '\n';

cin >> n;

switch (n) {

case 1:

work1();

break;

case 2:

work2();

break;

case 3:

work3();

break;

case 4:

work4();

break;

default:

cout << "Goodbye";

return 0;

break;

}

}