**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по МДК 01.01 Системное программирование

на тему

Разработка консольных приложений с использованием одномерных массивов. Методы для работы с одномерными массивами.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <math.h>

using namespace std;

iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

math.h — заголовочный файл стандартной библиотеки разработанный для выполнения простых математических операций.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

Все программы оформлены ввиде функций, т.е пример вызова 1 задания:

void f1()

{

int myArr[10];

const int length = sizeof(myArr) / sizeof(myArr[0]);

int sumResult = 0;

int max = 0, min = 0;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> myArr[i];

if (myArr[i] > max)

{

max = myArr[i];

}

if (myArr[i] < min)

{

min = myArr[i];

}

}

cout << "Минимальный элемент:\t" << min << endl;

cout << "Максимальный элемент:\t" << max << endl;

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

{

sumResult += myArr[i];

}

cout << "Сумма элементов массива:\t" << sumResult;

}

int main()

{

setlocale(LC\_ALL, "Russian");

f1();

}

Дана последовательность из 10 различных целых чисел. Найти сумму чисел этой последовательности, расположенных между максимальным и минимальным числами (в сумму включить и оба эти числа).

int myArr[10];

const int length = sizeof(myArr) / sizeof(myArr[0]);

int sumResult = 0;

int max = myArr[0], min = myArr[0];

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> myArr[i];

if (myArr[i] > max)

{

max = myArr[i];

}

if (myArr[i] < min)

{

min = myArr[i];

}

}

cout << "Минимальный элемент:\t" << min << endl;

cout << "Максимальный элемент:\t" << max << endl;

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

{

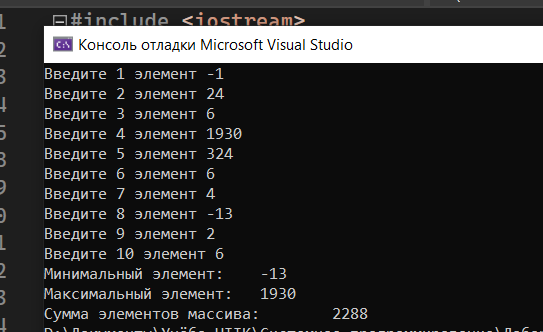
sumResult += myArr[i];

}

cout << "Сумма элементов массива:\t" << sumResult;

Из интересного расчёт длины массива:

const int length = sizeof(myArr) / sizeof(myArr[0]);



Даны координаты n точек на плоскости: х1, у1, ... , хn, уn (n=10), которые являются вершинами ломаной. Найти номера 2-х точек, расстояние между которыми наибольшее. Считать, что такая пара точек единственная.

const int n = 10;

double pointX[n], pointY[n];

int indexX, indexY;

double distanceOfPoints;

double ch = 0;

const int length = sizeof(pointX) / sizeof(pointX[0]);

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

{

cout << "Введите\t" << i + 1 << " элемент " << endl;

cout << "Точка X:\t";

cin >> pointX[i];

cout << "Точка Y:\t";

cin >> pointY[i];

}

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

{

for (int j = i + 1; j < length - 1; j++)

{

distanceOfPoints = sqrt(pow(pointX[i] - pointX[j], 2) + pow(pointY[i] - pointY[j], 2));

if (distanceOfPoints > ch)

{

ch = distanceOfPoints;

indexX = i;

indexY = j;

}

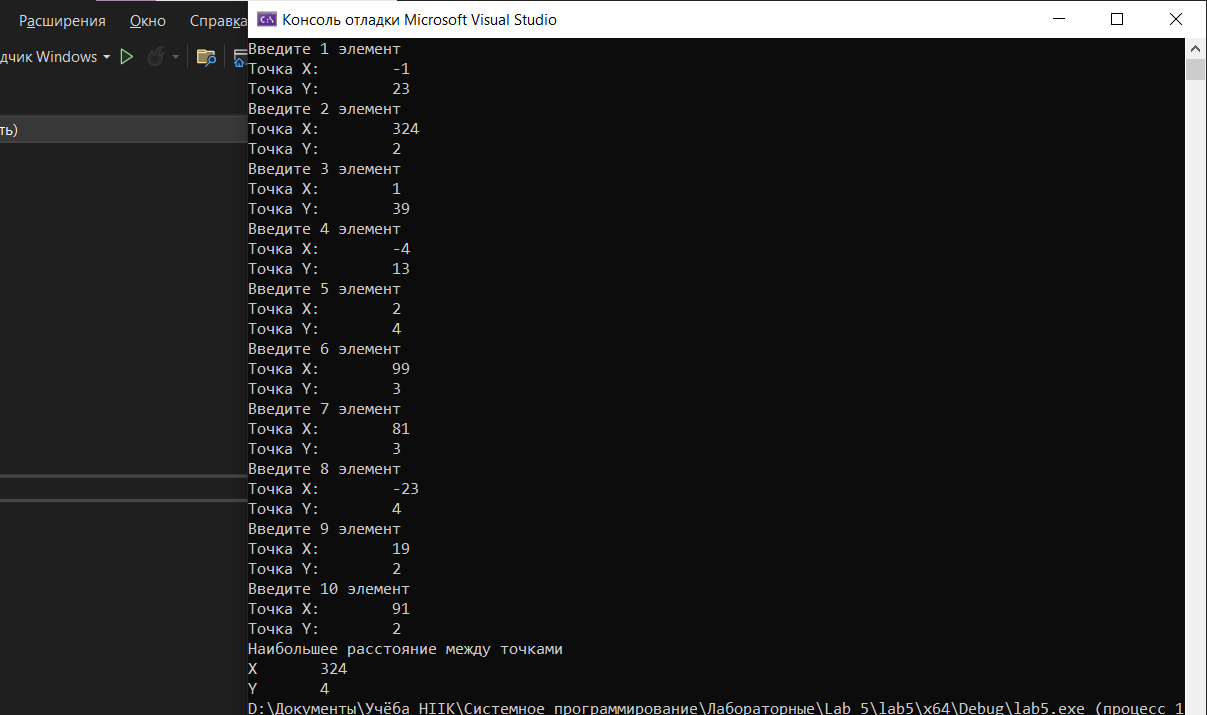
}

}

cout << "Наибольшее расстояние между точками" << endl;

cout << "X\t" << pointX[indexX] << endl;

cout << "Y\t" << pointY[indexY];



Дан целочисленный массив А[n]. Минимальный элемент этого массива заменить целой частью среднего арифметического всех элементов массива, остальные члены оставить без изменения.

int const n = 10;

double A[n];

const int length = sizeof(A) / sizeof(A[0]);

int sumResult = 0;

int indexMin;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

}

double min = A[0];

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

{

if (A[i] < min)

{

min = A[i];

indexMin = i;

}

}

cout << "Минимальный элемент:\t" << min << endl;

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

{

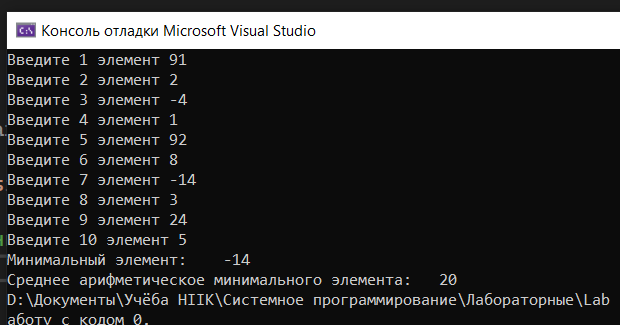
sumResult += A[i];

}

sumResult /= length;

A[indexMin] = sumResult;

cout << "Среднее арифметическое минимального элемента:\t" << A[indexMin];



Даны натуральные числа А1, ...,Аn. Определить количество членов Аk последовательности А1,...,Аn, удовлетворяющие условию Аk<(Ak-1 + Аk+1)/2.

const int n = 10;

double A[n], Ak[n];

const int length = sizeof(A) / sizeof(A[0]);

int counter = 0;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

Ak[i] = A[i];

}

for (int k = 1; k < length; k++)

{

if (Ak[k] < (Ak[k - 1] + A[k + 1]) / 2)

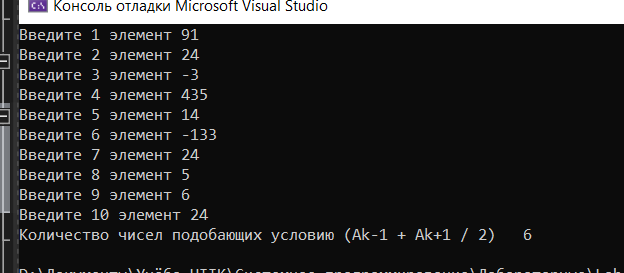
{

counter++;

}

}

cout << "Количество чисел подобающих условию (Ak-1 + Ak+1 / 2)\t" << counter << endl;



Даны натуральные числа А1, ...,Аn. Определить количество членов Аk последовательности А1, ...,Аn, удовлетворяющих условию Аk>2к.

const int n = 12;

double A[n], Ak[n];

const int length = sizeof(A) / sizeof(A[0]);

int counter = 0;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

Ak[i] = A[i];

}

for (int k = 1; k < length; k++)

{

if (Ak[k] > pow(2, k))

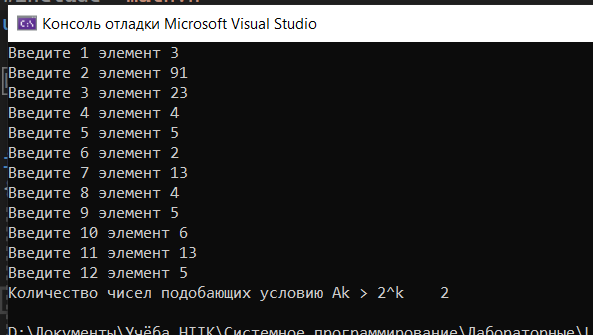
{

counter++;

}

}

cout << "Количество чисел подобающих условию Ak > 2^k\t" << counter << endl;



Даны натуральное число n и целые числа А1, ...,Аn. Найти наименьшее из четных чисел, входящих в последовательность А1,...,Аn.

const int n = 12;

int A[n];

const int length = sizeof(A) / sizeof(A[0]);

int counter = 0;

double min;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

}

min = A[0];

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

{

if (A[i] % 2 == 0)

{

if (A[i] < min && A[i] != 0)

{

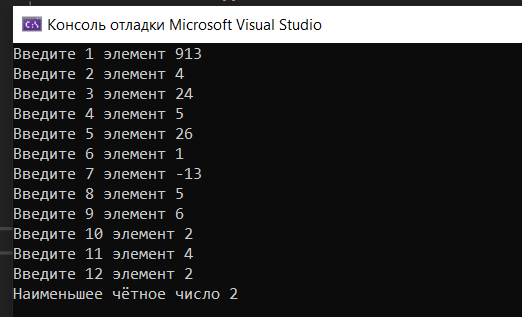
min = A[i];

}

}

}

cout << "Наименьшее чётное число\t" << min << endl;



Даны натуральные n и целые числа А1,...,Аn. Найти наибольшее из нечетных и количество четных чисел, входящих в последовательность А1,...,Аn.

const int n = 12;

int A[n];

const int length = sizeof(A) / sizeof(A[0]);

int counter = 0;

double max;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

}

max = A[0];

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

{

if (A[i] % 2 != 0)

{

if (A[i] > max && A[i] != 0)

{

max = A[i];

}

}

else if (A[i] % 2 == 0)

{

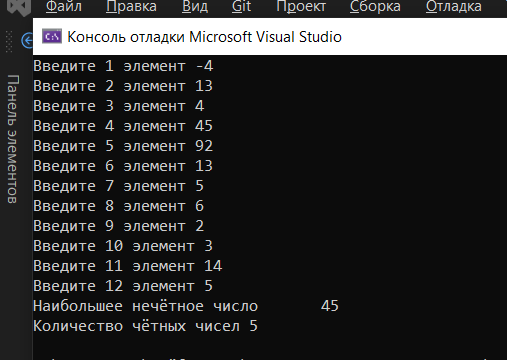
counter++;

}

}

cout << "Наибольшее нечётное число\t" << max << endl;

cout << "Количество чётных чисел\t" << counter << endl;



Дана последовательность из 15 целых чисел. Найти наибольшее и наименьшее число в этой последовательности и поменять эти числа местами.

const int n = 15;

int A[n];

const int length = sizeof(A) / sizeof(A[0]);

int counter = 0;

double max, min;

int maxIndex = 0, minIndex = 0;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

}

max = A[0];

min = A[0];

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

{

if (A[i] > max)

{

max = A[i];

maxIndex = i;

}

if (A[i] < min)

{

min = A[i];

minIndex = i;

}

}

cout << "Массив без изменений\n";

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

{

cout << A[i] << "\t";

}

cout << endl;

A[maxIndex] = min;

A[minIndex] = max;

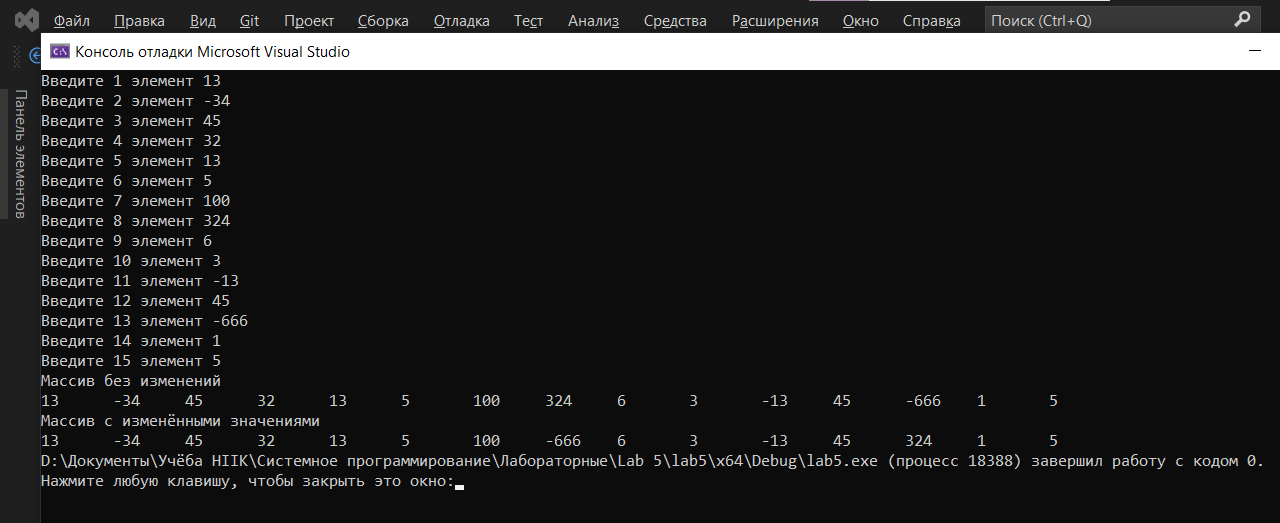
cout << "Массив с изменёнными значениями\n";

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

{

cout << A[i] << "\t";

}



Дана последовательность из 15 вещественных чисел. Наименьшее число в этой последовательности поменять местами с первым числом и наибольшее число поменять местами с последним числом в этой последовательности.

const int n = 15;

double A[n];

const int length = sizeof(A) / sizeof(A[0]);

int counter = 0;

double max, min;

int maxIndex = 0, minIndex = 0;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

}

max = A[0];

min = A[0];

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

{

if (A[i] > max)

{

max = A[i];

maxIndex = i;

}

if (A[i] < min)

{

min = A[i];

minIndex = i;

}

}

cout << "Массив без изменений\n";

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

{

cout << A[i] << "\t";

}

cout << endl;

A[maxIndex] = A[length - 1];

A[minIndex] = A[0];

A[0] = min;

A[length - 1] = max;

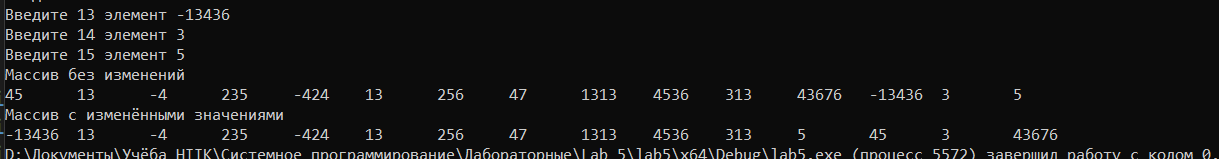
cout << "Массив с изменёнными значениями\n";

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

{

cout << A[i] << "\t";

}



Дан массив вещественных чисел, содержащий 10 элементов. Если минимальный элемент или максимальный элемент этого массива равен 0, то поменять эти элементы местами.

const int n = 10;

double A[n];

const int length = sizeof(A) / sizeof(A[0]);

double max, min;

int maxIndex = 0, minIndex = 0;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

}

max = A[0];

min = A[0];

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

{

if (A[i] > max)

{

max = A[i];

maxIndex = i;

}

if (A[i] < min)

{

min = A[i];

minIndex = i;

}

}

cout << "Массив без изменений\n";

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

{

cout << A[i] << "\t";

}

cout << endl;

if (A[maxIndex] == 0)

{

A[minIndex] = 0;

A[maxIndex] = min;

}

if (A[minIndex] == 0)

{

A[maxIndex] = 0;

A[minIndex] = max;

}

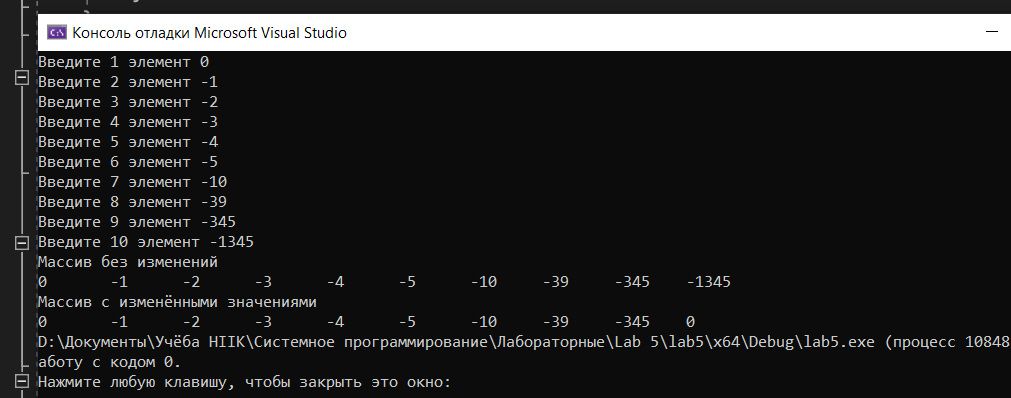
cout << "Массив с изменёнными значениями\n";

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

{

cout << A[i] << "\t";

}



Дан массив целых чисел, содержащий 10 элементов. Если минимальный элемент в этом массиве находится на первом месте, а максимальный – на последнем месте, то поменять местами эти элементы.

const int n = 10;

int A[n];

const int length = sizeof(A) / sizeof(A[0]);

double max, min;

int maxIndex = 0, minIndex = 0;

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

{

cout << "Введите\t" << i + 1 << " элемент ";

cin >> A[i];

}

max = A[0];

min = A[0];

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

{

if (A[i] > max)

{

max = A[i];

maxIndex = i;

}

if (A[i] < min)

{

min = A[i];

minIndex = i;

}

}

cout << "Массив без изменений\n";

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

{

cout << A[i] << "\t";

}

cout << endl;

if (A[0] == min && A[length - 1] == max)

{

A[maxIndex] = min;

A[minIndex] = max;

}

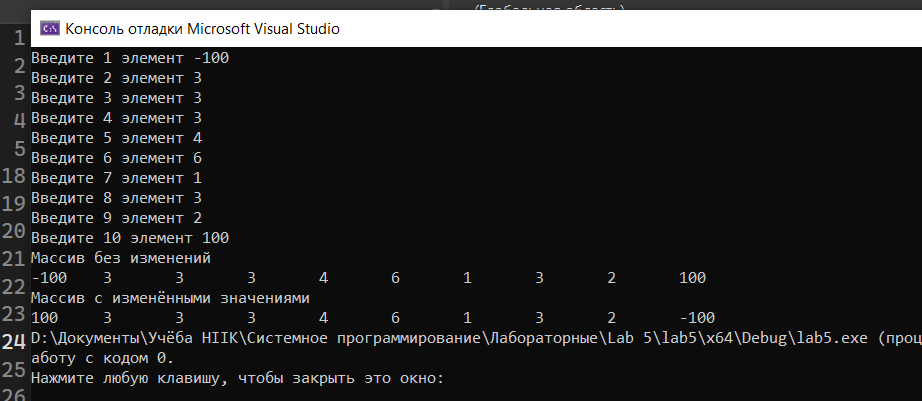
cout << "Массив с изменёнными значениями\n";

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

{

cout << A[i] << "\t";

}



**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

по МДК 01.01 Системное программирование

на тему

Разработка консольных приложений с использованием двумерных массивов. Методы для работы с двумерными массивами.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <cmath>

#include <cstdlib>

#include <ctime>

#include <algorithm>

#include <vector>

#include <iomanip>

using namespace std;

iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

cmath – заголовочный файл стандартной библиотеки разработанный для выполнения простых математических операций.

cstdlib – заголовочный файл, который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие.

ctime – включает в себя фунцкии времени.

algorithm – заголовочный файл в стандартной библиотеке языка программирования C++, включающий набор функций для выполнения алгоритмических операций.

vector – стандартный шаблон обобщённого программирования языка C++, реализующий динамический массив.

iomanip – библиотека, в которой содержится модификатор setw.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

Дана вещественная матрица размером NxM. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (один из них) оказал-ся в верхнем левом углу.

#include<iostream>

using namespace std;

int main()

{

int i, j, n, m;

double myArr[3][4], max, cur;

cout << "Write elemets 3 X 4:" << endl;

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

for (j = 0; j < 4; j++)

cin >> myArr[i][j];

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

{

for (int j = 0; j < 4; j++)

cout << myArr[i][j] << "\t";

cout << endl;

}

max = myArr[0][0];

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

for (j = 0; j < 4; j++)

if (myArr[i][j] > max)

{

max = myArr[i][j];

n = i, m = j;

}

cout << "Max = myArr" << "[" << n << "][" << m << "] = " << max << endl;

//обмен верхнего левого и найденного максимального элементов

cur = myArr[0][0];

myArr[0][0] = max;

myArr[n][m] = cur;

cout << "Result:\n";

for (int i = 0; i < 7; i++)

{

for (int j = 0; j < 4; j++)

cout << myArr[i][j] << "\t";

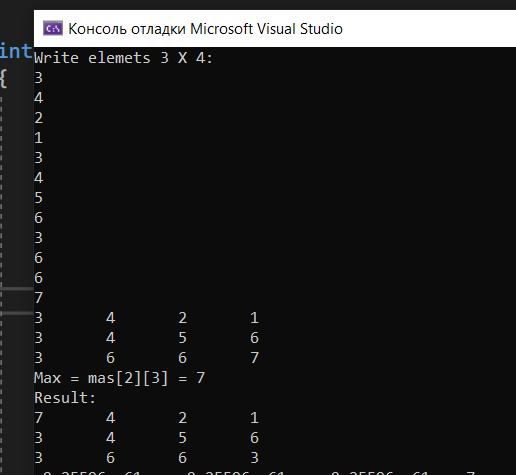
cout << endl;

}

system("pause");

return 0;

}



Дана вещественная матрица размером NxM. Упорядочить ее строки по возрастанию их первых элементов.

#include "iostream"

#include <iomanip>

using namespace std;

int main()

{

double\*\* a, b;

int i, j, n, m, k;

cout << "Vvedite razmer: n,m" << endl;

cin >> n >> m;

a = new double\* [n];

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

a[i] = new double[m];

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

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

{

cout << "Vvedite a[" << i << "][" << j << "]:";

cin >> a[i][j];

}

cout << "MASSIVE:" << endl;

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

{

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

cout << setw(10) << a[i][j] << " ";

cout << endl;

}

for (i = 0; i < n - 1; i++) {

for (k = 0; k < n - 1; k++) {

if (a[k][0] > a[k + 1][0]) {

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

b = a[k][j];

a[k][j] = a[k + 1][j];

a[k + 1][j] = b;

}

}

}

}

cout << "Result:" << endl;

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

{

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

cout << setw(10) << a[i][j] << " ";

cout << endl;

}

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

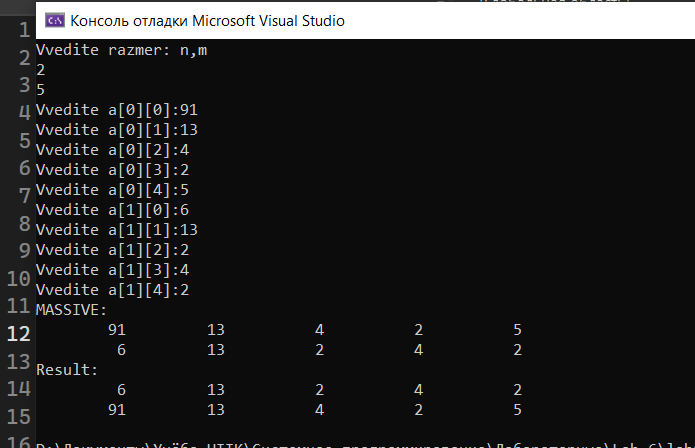
delete[] a[i];

delete[]a;

a = NULL;

return 0;

}



Дана вещественная матрица размером NхM. Упорядочить ее строки по возрастанию суммы их элементов.

#include <iostream>

#include <algorithm>

using namespace std;

bool compareRows(int\* a, int\* b, int m) {

int sumA = 0, sumB = 0;

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

sumA += a[i];

sumB += b[i];

}

return sumA < sumB;

}

int main() {

int n, m;

cout << "Vvedite razmer: n,m" << endl;

cin >> n >> m;

int\*\* myArr = new int\* [n];

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

myArr[i] = new int[m];

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

cout << "Vvedite a[" << i << "][" << j << "]:";

cin >> myArr[i][j];

}

}

sort(myArr, myArr + n, [m](int\* a, int\* b) { return compareRows(a, b, m); });

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

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

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

}

cout << endl;

}

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

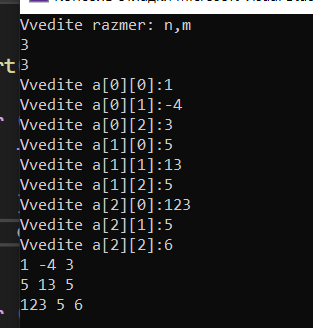
delete[] myArr[i];

}

delete[] myArr;

return 0;

}



Дана вещественная матрица размером NхM. Упорядочить ее строки по возрастанию наибольших элементов.

#include <iostream>

#include <algorithm>

using namespace std;

bool compareRows(int\* a, int\* b, int m) {

int maxA = a[0], maxB = b[0];

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

maxA = max(maxA, a[i]);

maxB = max(maxB, b[i]);

}

return maxA < maxB;

}

int main() {

int n, m;

cout << "Vvedite razmer: n,m" << endl;

cin >> n >> m;

int\*\* matrix = new int\* [n];

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

matrix[i] = new int[m];

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

cout << "Vvedite a[" << i << "][" << j << "]:";

cin >> matrix[i][j];

}

}

sort(matrix, matrix + n, [m](int\* a, int\* b) { return compareRows(a, b, m); });

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

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

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

}

cout << endl;

}

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

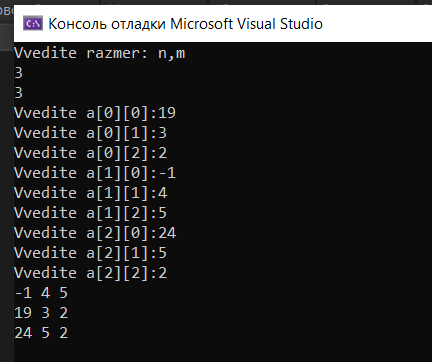
delete[] matrix[i];

}

delete[] matrix;

return 0;

}



Определить, является ли заданная целочисленная квадратная матрица по-рядка N симметричной относительно главной диагонали.

#include <iostream>

#include <vector>

using namespace std;

bool isSymmetric(vector<vector<int>> matrix, int n) {

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

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

if (matrix[i][j] != matrix[j][i]) {

return false;

}

}

}

return true;

}

int main() {

int n;

cout << "Vvedite razmer: n,n" << endl;

cin >> n;

vector<vector<int>> myArr(n, vector<int>(n));

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

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

cout << "Vvedite myArr[" << i << "][" << j << "]:";

cin >> myArr[i][j];

}

}

if (isSymmetric(myArr, n)) {

cout << "Matrix is symmetric" << endl;

}

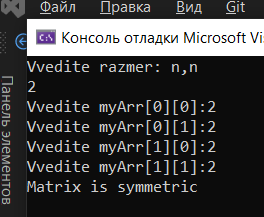
else {

cout << "Matrix is not symmetric" << endl;

}

return 0;

}



Среди тех строк целочисленной матрицы, которые содержат только нечетные элементы, найти строку с максимальной суммой модулей элементов.

#include <iostream>

#include <cmath>

#include <vector>

using namespace std;

int main() {

int n, m;

cout << "Vvedite razmer: n,m" << endl;

cin >> n >> m;

vector<vector<int>> matrix(n, vector<int>(m));

int max\_sum = 0;

int max\_sum\_index = 0;

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

int sum = 0;

bool only\_odd = true;

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

cout << "Vvedite myArr[" << i << "][" << j << "]:";

cin >> matrix[i][j];

if (matrix[i][j] % 2 == 0) {

only\_odd = false;

break;

}

sum += abs(matrix[i][j]);

}

if (only\_odd && sum > max\_sum) {

max\_sum = sum;

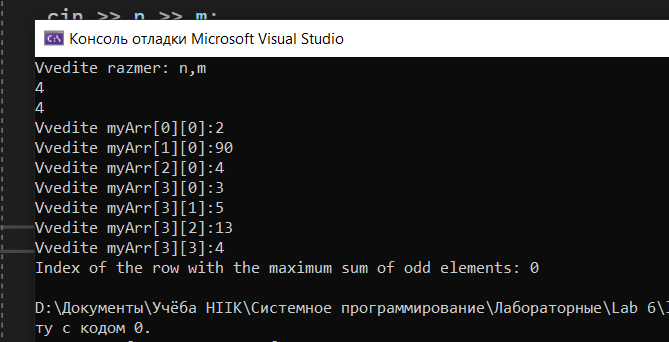
max\_sum\_index = i;

}

}

cout << "Index of the row with the maximum sum of odd elements: " << max\_sum\_index << endl;

return 0;

}

Среди тех столбцов целочисленной матрицы, которые содержат только такие элементы, значения которых по модулю не превышают 10, найти столбец с минимальным произведением элементов.

#include <iostream>

#include <climits>

using namespace std;

const int N = 100;

int myArr[N][N];

int main() {

int m, n;

cout << "Vvedite razmer: n,m" << endl;

cin >> m >> n;

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

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

cout << "Vvedite myArr[" << i << "][" << j << "]:";

cin >> myArr[i][j];

}

int min\_col = INT\_MAX;

int min\_product = INT\_MAX;

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

int product = 1;

bool flag = true;

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

if (abs(myArr[i][j]) > 10) {

flag = false;

break;

}

product \*= myArr[i][j];

}

if (flag) {

if (product < min\_product) {

min\_product = product;

min\_col = j;

}

}

}

if (min\_col == INT\_MAX) {

cout << "No such column found" << endl;

}

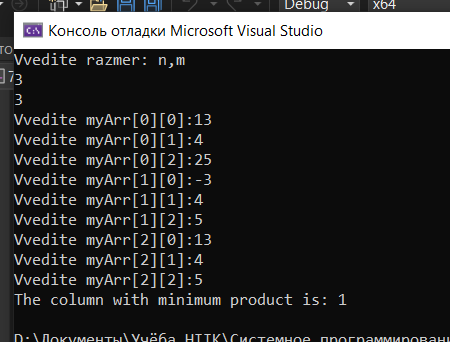
else {

cout << "The column with minimum product is: " << min\_col << endl;

}

return 0;

}



Даны целые числа А1,...,А10, целочисленная квадратная матрица порядка N. Заменить нулями в матрице те элементы, для которых имеются равные числа среди А1,...,А10.

#include <iostream>

using namespace std;

int main()

{

const int N = 20;

int myArr[N][N] = { { 0 } };

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

{

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

{

myArr[row][col] = rand() % 9 + 1;

cout << myArr[row][col] << " ";

}

cout << endl;

}

cout << endl;

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

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

if (((row + col) % 2) == 0)

{

int r\_fwd = 0, c\_fwd = 0;

while (r\_fwd < N)

{

if (c\_fwd >= N) { c\_fwd = 0; r\_fwd++; }

if (myArr[r\_fwd][c\_fwd] == myArr[row][col] && r\_fwd != row && c\_fwd != col)

myArr[r\_fwd][c\_fwd] = 0;

c\_fwd++;

}

}

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

{

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

cout << myArr[row][col] << " ";

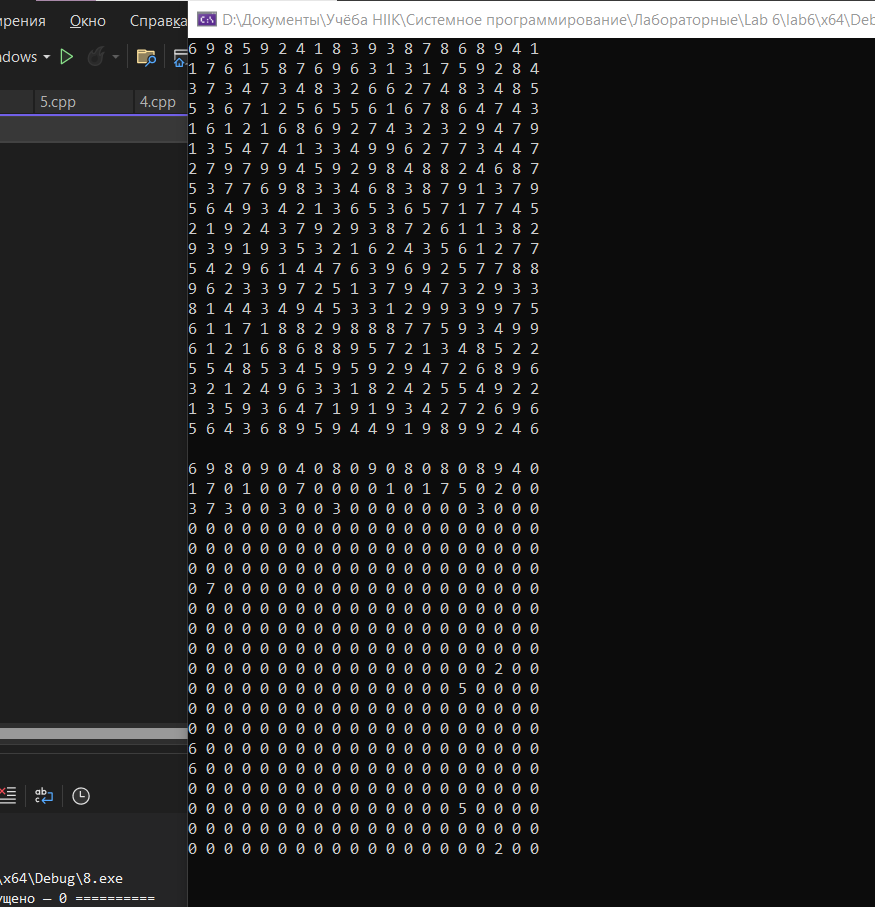
cout << endl;

}

cout << endl << endl;

cin.get();

}



В двумерном массиве целых чисел поменять местами строки, симметричные относительно середины массива (горизонтальной линии).

#include <iostream>

#include <algorithm>

using namespace std;

const int N = 10;

void swap(int a[], int b[], int size) {

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

swap(a[i], b[i]);

}

}

int main() {

int myArr[N][N];

int n, m;

cout << "Vvedite razmer: n,m" << endl;

cin >> n >> m;

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

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

cout << "Vvedite myArr[" << i << "][" << j << "]:";

cin >> myArr[i][j];

}

}

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

swap(myArr[i], myArr[n - i - 1], m);

}

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

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

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

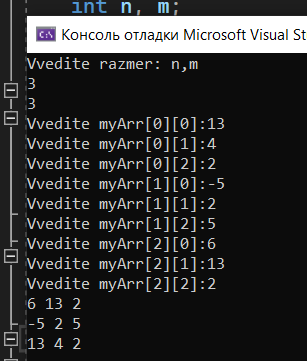
}

cout << endl;

}

return 0;

}



В двумерном массиве целых чисел поменять местами столбцы, симмет-ричные относительно середины массива (вертикальной линии).

#include<iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main()

{

int myArr[100][100], i, j, t, n, m;

cout << "n= "; cin >> n;

cout << "m= "; cin >> m;

cout << "Matr:" << endl;

srand(time(NULL));

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

{

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

{

myArr[i][j] = rand() % 10;

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

}

cout << endl;

}

for (j = 0; j < m / 2; j++)

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

{

t = myArr[i][j]; myArr[i][j] = myArr[i][m - 1 - j]; myArr[i][m - 1 - j] = t;

}

cout << endl << "New matr:" << endl;

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

{

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

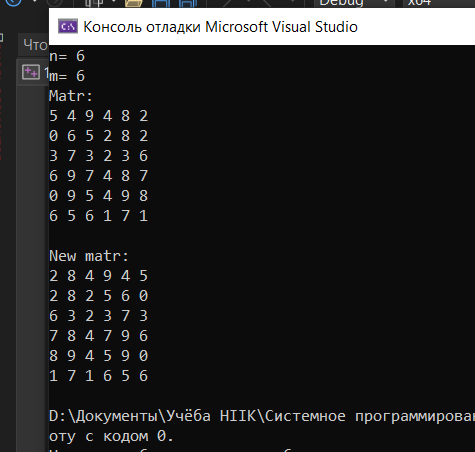
cout << myArr[i][j] << " ";

cout << endl;

}

return 0;

}



Даны две действительные квадратные матрицы порядка N. Получить новую матрицу прибавлением к элементам каждого столбца первой матрицы минимального элемента соответствующего столбца второй матрицы.

#include <iostream>

#include <cmath>

using namespace std;

const int N = 5;

void add\_min\_col(double a[][N], double b[][N], double c[][N]) {

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

double min\_val = b[0][j];

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

if (b[i][j] < min\_val) {

min\_val = b[i][j];

}

}

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

c[i][j] = a[i][j] + min\_val;

}

}

}

int main() {

double a[N][N] = { {1, 2, 3, 4, 5},

{6, 7, 8, 9, 10},

{11, 12, 13, 14, 15},

{16, 17, 18, 19, 20},

{21, 22, 23, 24, 25} };

double b[N][N] = { {25, 24, 23, 22, 21},

{20, 19, 18, 17, 16},

{15, 14, 13, 12, 11},

{10, 9, 8, 7, 6},

{5, 4, 3, 2, 1} };

double c[N][N];

add\_min\_col(a, b, c);

cout << "Result matrix:" << endl;

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

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

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

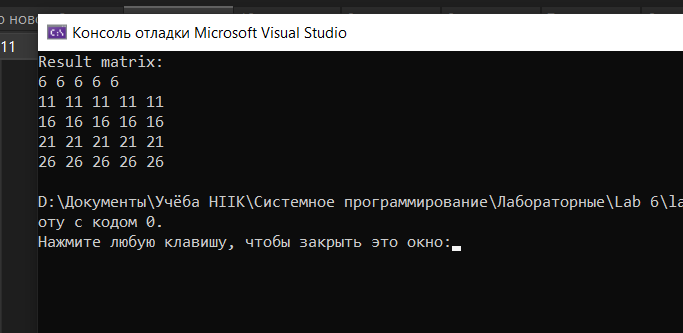
}

cout << endl;

}

return 0;

}



**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

по МДК 01.01 Системное программирование

на тему

Строки. Работа со строками. Методы для работы со строками.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <sstream>

#include <string>

#include <algorithm>

#include <vector>

#include <iomanip>

using namespace std;

iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

algorithm – заголовочный файл в стандартной библиотеке языка программирования C++, включающий набор функций для выполнения алгоритмических операций.

string – заголовочный файл в стандартной библиотеке языка С++ содержаший тип данных строки и набор функций для работы с ней.

sstream – заголовочный файл с классами, функциями и переменными для организации работы со строками, через интерфейс потоков, в языке программирования C++.

vector – стандартный шаблон обобщённого программирования языка C++, реализующий динамический массив.

iomanip – библиотека, в которой содержится модификатор setw.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

Дан текст. а) Подсчитать количество слов в данной строке.

#include <iostream>

#include <sstream>

int main() {

std::string text = "My own text Medvedev";

std::istringstream iss(text);

std::string word;

int word\_count = 0;

while (iss >> word) {

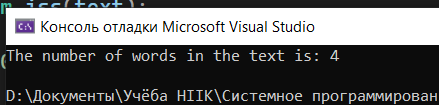
++word\_count;

}

std::cout << "The number of words in the text is: " << word\_count << std::endl;

return 0;

}



Подсчитать количество букв а в последнем слове данной строки.

#include <iostream>

#include <string>

using namespace std;

int main() {

string str;

getline(cin, str);

int count = 0;

int len = str.length();

for (int i = len - 1; i >= 0; i--) {

if (str[i] == ' ') {

break;

}

else if (str[i] == 'a') {

count++;

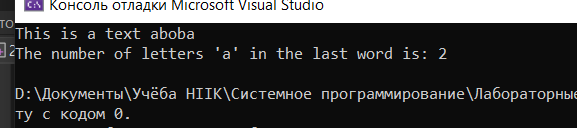
}

}

cout << "The number of letters 'a' in the last word is: " << count << endl;

return 0;

}



Найти количество слов, начинающихся с буквы б.

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

int count = 0;

while (ss >> word) {

if (word[0] == 'b' || word[0] == 'B') {

count++;

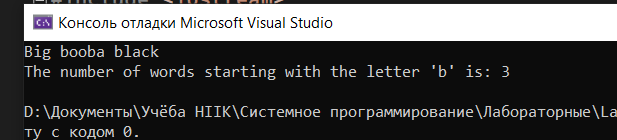
}

}

cout << "The number of words starting with the letter 'b' is: " << count << endl;

return 0;

}



Найти количество слов, у которых первый и последний символы совпадают между собой.

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

int count = 0;

while (ss >> word) {

if (word[0] == word[word.length() - 1]) {

count++;

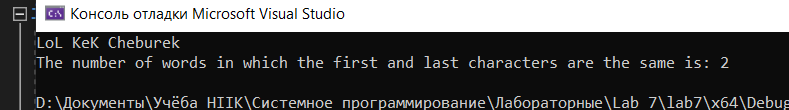
}

}

cout << "The number of words in which the first and last characters are the same is: " << count << endl;

return 0;

}



Найти длину самого короткого слова.

#include <iostream>

#include <string>

#include <sstream>

#include <climits>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

int minLength = INT\_MAX;

while (ss >> word) {

if (word.length() < minLength) {

minLength = word.length();

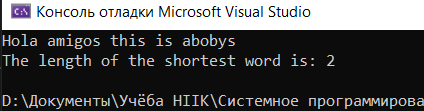
}

}

cout << "The length of the shortest word is: " << minLength << endl;

return 0;

}



Составить программу циклической перестановки букв в словах текста так, что i-я буква слова становится i+1-ой, а последняя - первой.

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

while (ss >> word) {

int length = word.length();

char first = word[length - 1];

for (int i = length - 1; i > 0; i--) {

word[i] = word[i - 1];

}

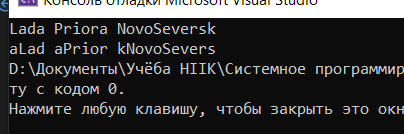
word[0] = first;

cout << word << " ";

}

return 0;

}



В каждом слове текста замените "а" на букву "е", если "а" стоит на четном месте, и заменить букву "б" на сочетание "ак", если "б" стоит на нечетном месте.

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

while (ss >> word) {

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

if (word[i] == 'a' && i % 2 == 0) {

word[i] = 'e';

}

else if (word[i] == 'b' && i % 2 != 0) {

word.replace(i, 1, "ak");

i += 1;

}

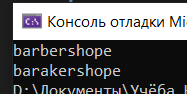
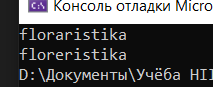
}

cout << word << " ";

}

return 0;

}

Дан текст, содержащий от 2 до 30 слов, в каждом из которых от 2 до 10 латинских букв; между соседними словами – не менее одного пробела. Напечатать все слова, отличные от последнего слова, предварительно преобразовав каждое из них по следующему правилу: 1) перенести первую букву в конец слова; 2) перенести последнюю букву в начало слова.

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

int count = 0;

while (ss >> word) {

count++;

if (count == 1) {

continue;

}

char first = word[0];

char last = word[word.length() - 1];

word.erase(0, 1);

word.erase(word.length() - 1, 1);

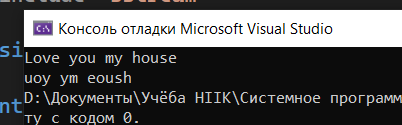
word = last + word + first;

cout << word << " ";

}

return 0;

}



Дан текст. Напечатать все слова, отличные от последнего слова, предварительно преобразовав каждое из них по следующему правилу: 1) оставить в слове только первые вхождения каждой буквы; 2) если слово нечетной длины, то удалить его среднюю букву

#include <iostream>

#include <string>

#include <sstream>

#include <unordered\_set>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

int count = 0;

while (ss >> word) {

count++;

if (count == 1) {

continue;

}

unordered\_set<char> seen;

string result = "";

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

if (seen.count(word[i]) == 0) {

result += word[i];

seen.insert(word[i]);

}

}

if (result.length() % 2 == 1) {

result.erase(result.length() / 2, 1);

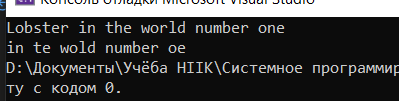
}

cout << result << " ";

}

return 0;

}



Составить таблицу слов данного текста, начинающихся с буквы "*А*", с указанием числа повторений каждого слова.

#include <iostream>

#include <string>

#include <sstream>

#include <map>

using namespace std;

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

map<string, int> table;

while (ss >> word) {

if (word[0] == 'A' || word[0] == 'a') {

table[word]++;

}

}

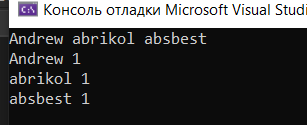
for (auto entry : table) {

cout << entry.first << " " << entry.second << endl;

}

return 0;

}



Составить программу для вычеркивания из слов текста всех букв, стоящих на нечетных местах после буквы "а".

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

void crossOutLetters(string& word);

void crossOutLetters(string& word) {

bool foundA = false;

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

if (foundA && i % 2 == 1) {

word[i] = '-';

}

if (word[i] == 'a' || word[i] == 'A') {

foundA = true;

}

}

}

int main() {

string str;

getline(cin, str);

stringstream ss(str);

string word;

while (ss >> word) {

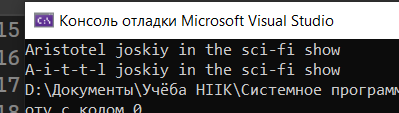
crossOutLetters(word);

cout << word << " ";

}

return 0;

}



12. Составить программы для перевода арабских чисел в римские и для обратной операции. Например, 255 = CCLV = сто + сто + пятьдесят + пять Замечание. Подобными алгоритмами перевода чисел из одной системы в другую мы пользуемся по нескольку раз на дню, когда ведем денежные расчеты. Сумма денег – это арабское число, которому соответствует определенный набор банкнот и монет (аналоги римских цифр).

#include <iostream>

#include <string>

#include <map>

using namespace std;

string intToRoman(int num) {

map<int, string> values = {

{1000, "M"}, {900, "CM"}, {500, "D"}, {400, "CD"},

{100, "C"}, {90, "XC"}, {50, "L"}, {40, "XL"},

{10, "X"}, {9, "IX"}, {5, "V"}, {4, "IV"}, {1, "I"}

};

string result;

for (auto i = values.rbegin(); i != values.rend(); i++) {

while (num >= i->first) {

result += i->second;

num -= i->first;

}

}

return result;

}

int main() {

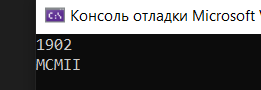
int num;

cin >> num;

cout << intToRoman(num) << endl;

return 0;

}



Подсчитать, сколько букв надо исправить в слове Х, чтобы получилось слово Y (Х,Y – слова одинаковой длины).

#include <iostream>

#include <string>

int main() {

std::string X, Y;

std::cin >> X >> Y;

int count = 0;

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

if (X[i] != Y[i]) {

count++;

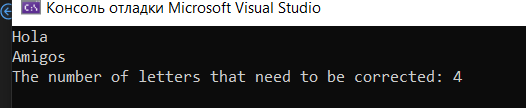
}

}

std::cout << "The number of letters that need to be corrected: " << count << std::endl;

return 0;

}



Составить программу для подсчета числа одинаковых букв в словах X и Y равной длины, стоящих на одних и тех же местах.

#include <iostream>

#include <string>

int main() {

std::string X, Y;

std::cin >> X >> Y;

int count = 0;

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

if (X[i] == Y[i]) {

count++;

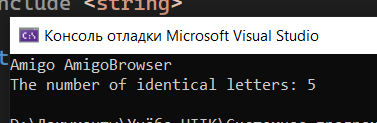
}

}

std::cout << "The number of identical letters: " << count << std::endl;

return 0;

}



15. Задано определенное количество конкретных сочетаний букв (например, УЩ, ЮЩ и др.). Определить, сколько таких групп символов содержится в тексте, вводимом с клавиатуры.

#include <iostream>

#include <string>

#include <vector>

int main() {

std::vector<std::string> combinations = { "USCH", "YUSCH", "..." };

std::string text;

std::cin >> text;

int count = 0;

for (const auto& combination : combinations) {

size\_t pos = text.find(combination);

while (pos != std::string::npos) {

count++;

pos = text.find(combination, pos + 1);

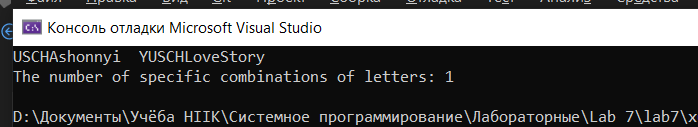
}

}

std::cout << "The number of specific combinations of letters: " << count << std::endl;

return 0;

}



С клавиатуры вводится текст. Подсчитать и вывести на печать количество слов текста, начинающихся с гласной.

#include <iostream>

#include <string>

int main() {

std::string text;

std::cin >> text;

int count = 0;

for (size\_t i = 0; i < text.length(); i++) {

if (text[i] == ' ') {

continue;

}

if (text[i] == 'a' || text[i] == 'e' || text[i] == 'i' || text[i] == 'o' || text[i] == 'u' || text[i] == 'A' || text[i] == 'E' || text[i] == 'I' || text[i] == 'O' || text[i] == 'U') {

count++;

}

while (i < text.length() && text[i] != ' ') {

i++;

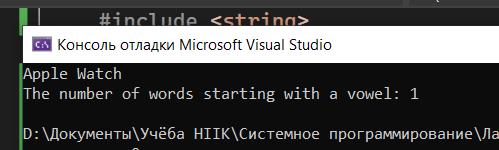
}

}

std::cout << "The number of words starting with a vowel: " << count << std::endl;

return 0;

}



Для заданного текста определить длину содержащейся в нем максимальной серии символов, отличных от латинских букв.

#include <iostream>

#include <string>

using namespace std;

int main() {

string text;

getline(cin, text);

int maxLength = 0;

int currentLength = 0;

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

char c = text[i];

if (!isalpha(c)) {

currentLength++;

if (currentLength > maxLength) {

maxLength = currentLength;

}

}

else {

currentLength = 0;

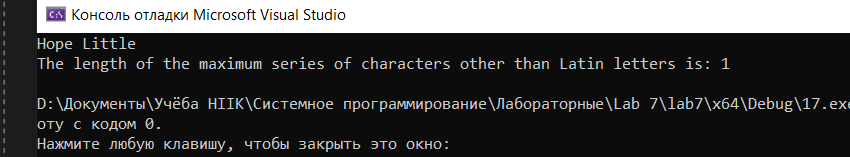
}

}

cout << "The length of the maximum series of characters other than Latin letters is: " << maxLength << endl;

return 0;

}



Записать программу, выясняющую, можно ли из букв слова X составить слово Y.

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

bool canForm(string x, string y) {

sort(x.begin(), x.end());

sort(y.begin(), y.end());

return x == y;

}

int main() {

string x, y;

cin >> x >> y;

if (canForm(x, y)) {

cout << "Yes, the letters of word X can form word Y." << endl;

}

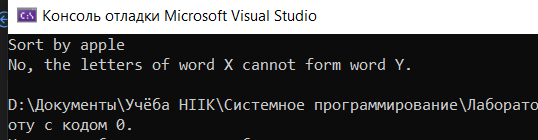
else {

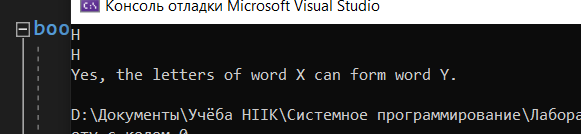
cout << "No, the letters of word X cannot form word Y." << endl;

}

return 0;

}





**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №10

по МДК 01.01 Системное программирование

на тему

Файлы. Работа с файлами.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <fstream>

#include <algorithm>

#include <vector>

#include <iomanip>

using namespace std;

iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

fstream – заголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл.

algorithm – заголовочный файл в стандартной библиотеке языка программирования C++, включающий набор функций для выполнения алгоритмических операций.

vector – стандартный шаблон обобщённого программирования языка C++, реализующий динамический массив.

iomanip – библиотека, в которой содержится модификатор setw.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

Дано имя файла и целое число N (> 1). Создать файл целых чисел с данным именем и записать в него N первых положительных четных чисел(2, 4, …).

#include <iostream>

#include <fstream>

using namespace std;

int main() {

int n;

string filename;

cout << "Enter the name of the file: ";

cin >> filename;

cout << "Enter a positive integer N: ";

cin >> n;

ofstream file;

file.open(filename);

for (int i = 2; i <= 2 \* n; i += 2) {

file << i << endl;

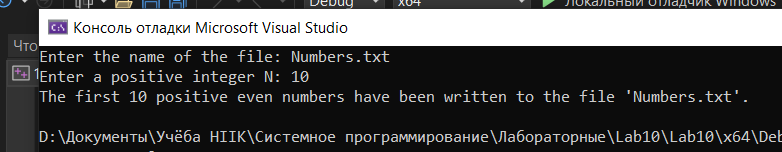
}

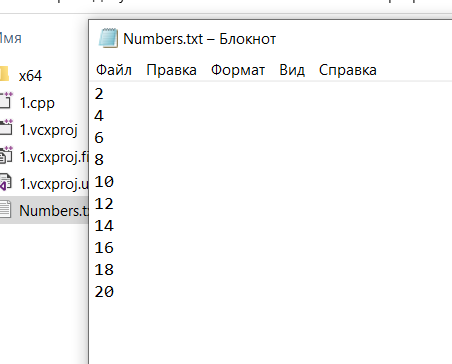
file.close();

cout << "The first " << n << " positive even numbers have been written to the file '" << filename << "'." << endl;

return 0;

}





Дано имя файла и вещественные числа A и D. Создать файл вещественных чисел с данным именем и записать в него 10 первых членов арифметической прогрессии с начальным членом A и разностью D : A, A + D, A + 2·D, A + 3·D, …

#include <iostream>

#include <fstream>

using namespace std;

int main() {

string filename;

float A, D;

cout << "Enter the name of the file: ";

cin >> filename;

cout << "Enter A: ";

cin >> A;

cout << "Enter D: ";

cin >> D;

ofstream file(filename, ios::out);

if (!file) {

cout << "Cannot open file" << endl;

return 1;

}

for (int i = 0; i < 10; i++) {

file << A + i \* D << endl;

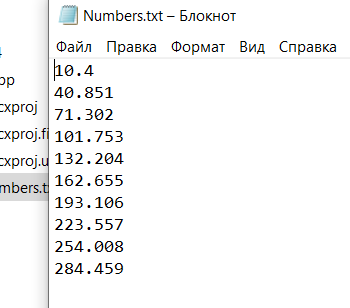
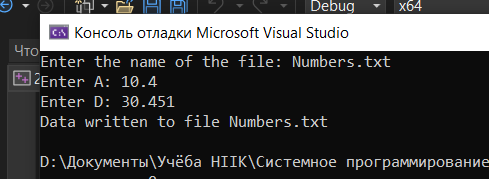
}

file.close();

cout << "Data written to file " << filename << endl;

return 0;

}



Дано имя файла целых чисел. Найти количество элементов, содержащихся в данном файле.Если файла с таким именем не существует, то вывести −1.

#include <fstream>

#include <iostream>

using namespace std;

int main() {

int kont = 0;

int x;

string filename;

cout << "Enter the name of the file: ";

cin >> filename;

ifstream fin(filename);

if (!fin) {

cout << -1;

return 0;

}

while (fin >> x) {

kont++;

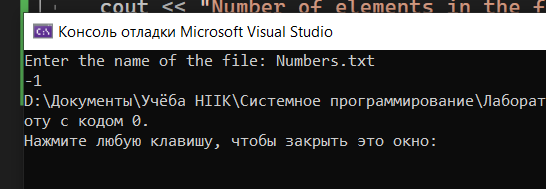
}

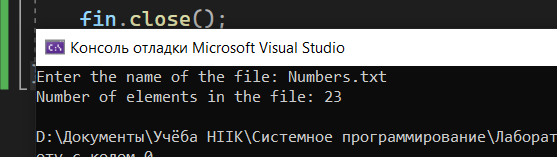
cout << "Number of elements in the file: " << kont << endl;

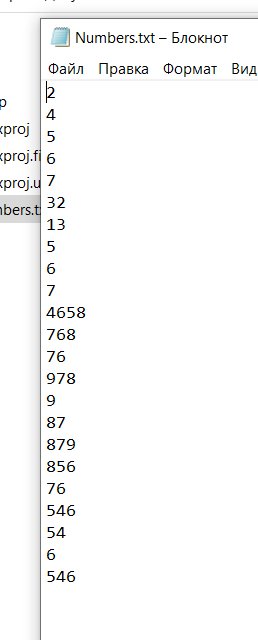
fin.close();

return 0;

}







Дано целое число K и файл, содержащий неотрицательные целые числа. Вывести K - й элемент файла.Если такой элемент отсутствует, то вывести −1.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

int k, num, cnt = 0;

cout << "Enter the value of k: ";

cin >> k;

string fileName;

cout << "Enter the name of the file: ";

cin >> fileName;

ifstream inFile(fileName);

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

cout << -1 << endl;

return 0;

}

while (inFile >> num) {

cnt++;

if (cnt == k) {

cout << num << endl;

return 0;

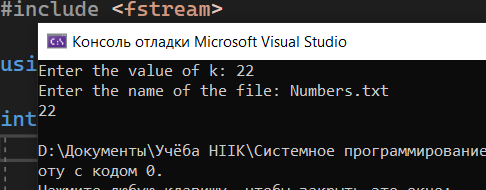
}

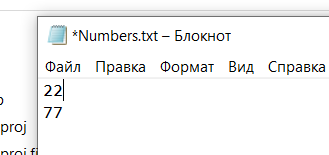
}

cout << -1 << endl;

return 0;

}





Дан файл целых чисел. Создать новый файл, содержащий те же элементы, что и исходный файл, но в обратном порядке.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

string in\_file\_name, out\_file\_name;

cout << "Enter the name of the input file: ";

cin >> in\_file\_name;

cout << "Enter the name of the output file: ";

cin >> out\_file\_name;

ifstream in\_file(in\_file\_name);

ofstream out\_file(out\_file\_name);

if (!in\_file.is\_open()) {

cout << "Error opening input file." << endl;

return 1;

}

vector<int> numbers;

int number;

while (in\_file >> number) {

numbers.push\_back(number);

}

if (!out\_file.is\_open()) {

cout << "Error opening output file." << endl;

return 1;

}

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

out\_file << numbers[i] << endl;

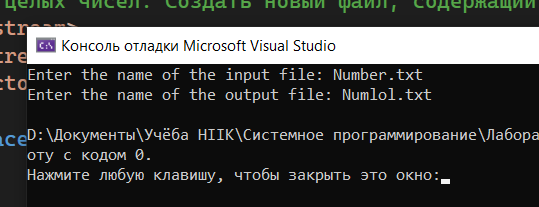
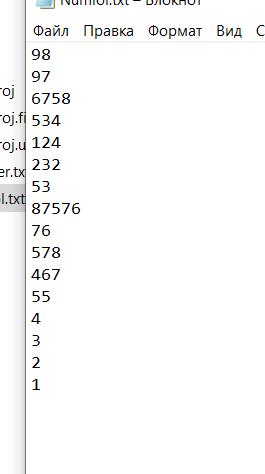
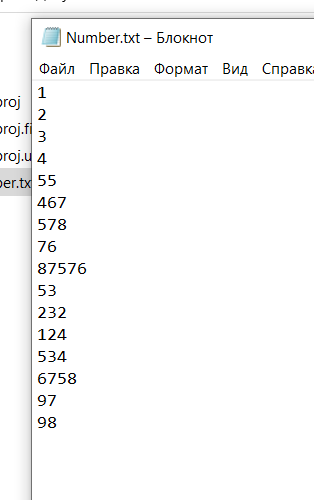
}

in\_file.close();

out\_file.close();

return 0;

}



Дан файл вещественных чисел. Создать два новых файла, первый из которых содержит элементы исходного файла с нечетными номерами(1, 3, …), а второй — с четными(2, 4, …).

#include <iostream>

#include <fstream>

using namespace std;

int main() {

string inputFileName, oddFileName, evenFileName;

cout << "Enter the name of the input file: ";

cin >> inputFileName;

cout << "Enter the name of the first output file (for odd numbers): ";

cin >> oddFileName;

cout << "Enter the name of the second output file (for even numbers): ";

cin >> evenFileName;

ifstream inputFile(inputFileName);

if (!inputFile) {

cout << "Error opening input file" << endl;

return 1;

}

ofstream oddFile(oddFileName);

ofstream evenFile(evenFileName);

int count = 0;

double number;

while (inputFile >> number) {

if (count % 2 == 0) {

evenFile << number << endl;

}

else {

oddFile << number << endl;

}

count++;

}

inputFile.close();

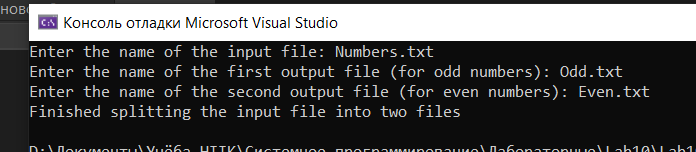
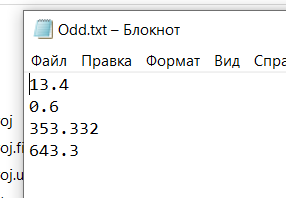
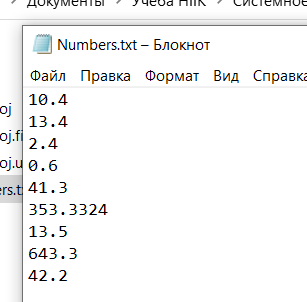
oddFile.close();

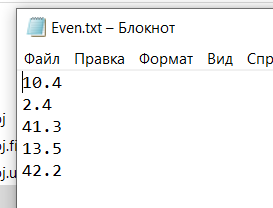
evenFile.close();

cout << "Finished splitting the input file into two files" << endl;

return 0;

}





Дан файл целых чисел. Создать два новых файла, первый из которых содержит положительные числа из исходного файла(в обратном порядке), а второй — отрицательные(также в обратном порядке). Если положительные или отрицательные числа в исходном файле отсутствуют, то соответствующий результирующий файл оставить пустым.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

string fileName;

cout << "Enter the name of the file: ";

cin >> fileName;

ifstream in(fileName);

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

cout << "-1" << endl;

return 0;

}

vector<int> positiveNumbers, negativeNumbers;

int number;

while (in >> number) {

if (number > 0) {

positiveNumbers.push\_back(number);

}

else if (number < 0) {

negativeNumbers.push\_back(number);

}

}

in.close();

ofstream outPositive(fileName + "\_positive.txt");

if (!positiveNumbers.empty()) {

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

outPositive << positiveNumbers[i] << endl;

}

}

outPositive.close();

ofstream outNegative(fileName + "\_negative.txt");

if (!negativeNumbers.empty()) {

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

outNegative << negativeNumbers[i] << endl;

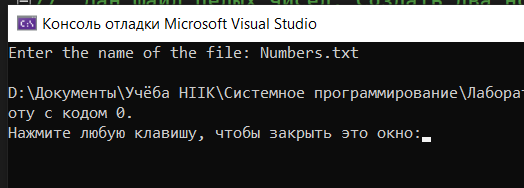
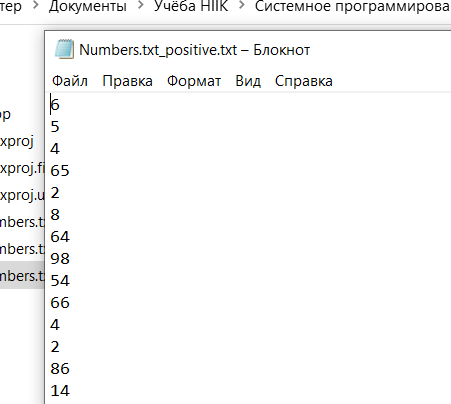
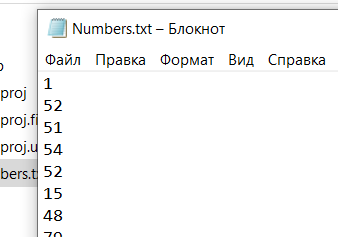
}

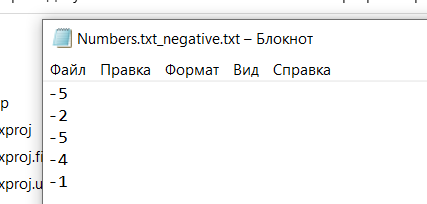
}

outNegative.close();

return 0;

}





Дан файл вещественных чисел. Найти среднее арифметическое его элементов.

#include <fstream>

#include <iostream>

using namespace std;

int main() {

double sum = 0, number, count = 0;

ifstream inFile("Numbers.txt"); // открытие файла для чтения

if (!inFile) { // проверка на успешное открытие файла

cout << "Could not open the file" << endl;

return 1;

}

while (inFile >> number) { // чтение всех чисел из файла

sum += number;

count++;

}

inFile.close(); // закрытие файла

if (count == 0) { // проверка на пустой файл

cout << "The file is empty" << endl;

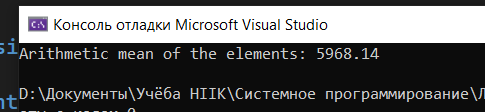
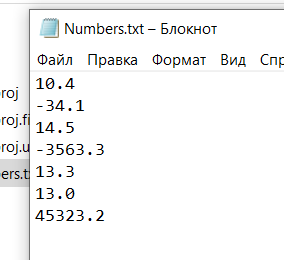
return 1;

}

cout << "Arithmetic mean of the elements: " << sum / count << endl;

return 0;

}



Дан файл вещественных чисел. Найти сумму его элементов с четными номерами.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ifstream inputFile("Numbers.txt");

double sum = 0.0, num;

int counter = 0;

while (inputFile >> num) {

if (counter % 2 == 0) {

sum += num;

}

counter++;

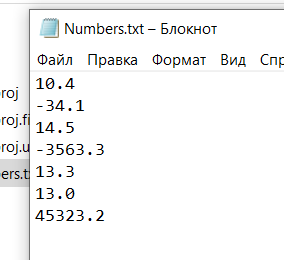
}

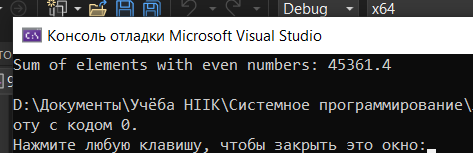
inputFile.close();

cout << "Sum of elements with even numbers: " << sum << endl;

return 0;

}





Дан файл вещественных чисел. Найти его первый локальный минимум (локальным минимумом называется элемент, который меньше своих соседей).

#include <fstream>

#include <iostream>

int main() {

std::ifstream input\_file("Numbers.txt");

double prev, curr, next;

if (input\_file >> prev) {

while (input\_file >> curr) {

if (input\_file >> next) {

if (prev > curr && curr < next) {

std::cout << "The first local minimum is " << curr << std::endl;

return 0;

}

prev = curr;

}

else {

break;

}

}

std::cout << "No local minimum found." << std::endl;

}

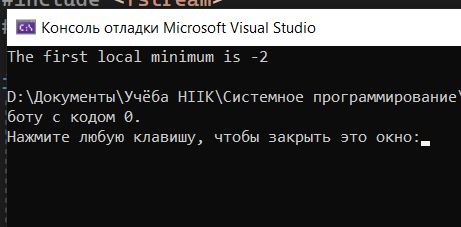
else {

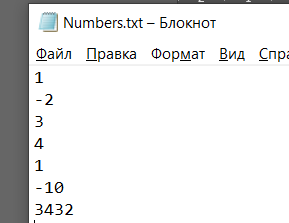
std::cout << "Input file is empty." << std::endl;

}

return 0;

}





Дан файл вещественных чисел. Поменять в нем местами минимальный и максимальный элементы.

#include <iostream>

#include <fstream>

#include <algorithm>

using namespace std;

int main() {

ifstream input\_file("Numbers.txt");

ofstream output\_file("Num.txt");

double num, min\_val, max\_val;

int min\_pos, max\_pos;

input\_file >> num;

min\_val = max\_val = num;

min\_pos = max\_pos = 0;

int pos = 0;

while (input\_file >> num) {

++pos;

if (num < min\_val) {

min\_val = num;

min\_pos = pos;

}

if (num > max\_val) {

max\_val = num;

max\_pos = pos;

}

}

input\_file.close();

input\_file.open("Numbers.txt");

pos = 0;

while (input\_file >> num) {

if (pos == min\_pos) {

output\_file << max\_val << endl;

}

else if (pos == max\_pos) {

output\_file << min\_val << endl;

}

else {

output\_file << num << endl;

}

++pos;

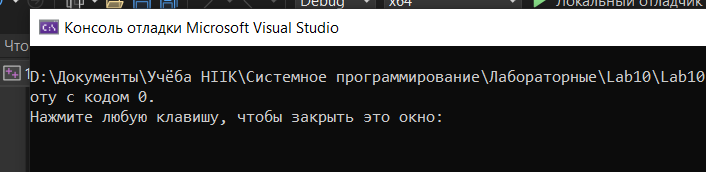
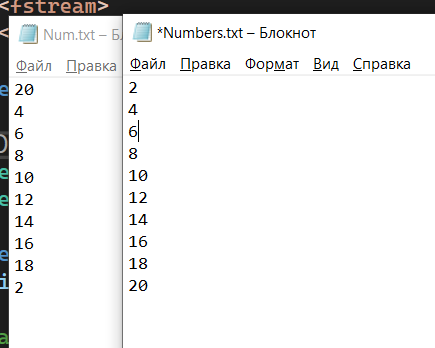
}

input\_file.close();

output\_file.close();

return 0;

}



Дан файл целых чисел с элементами A1, A2, …, AN (N — количество элементов в файле).Заменить исходное расположение его элементов на следующее : A1, AN, A2, AN−1, A3, … .

#include <iostream>

#include <fstream>

#include <vector>

int main() {

std::ifstream in("Number.txt");

std::ofstream out("Num.txt");

std::vector<int> arr;

int n;

in >> n;

int x;

while (in >> x) {

arr.push\_back(x);

}

for (int i = 0; i < arr.size() / 2; i++) {

out << arr[i] << " " << arr[arr.size() - i - 1] << " ";

}

if (arr.size() % 2 == 1) {

out << arr[arr.size() / 2];

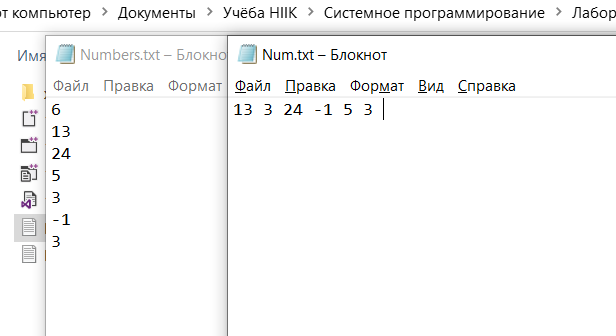
}

in.close();

out.close();

return 0;

}



Дан файл вещественных чисел. Заменить в файле каждый элемент, кроме начального и конечного, на его среднее арифметическое с предыдущим и последующим элементом.

#include <fstream>

#include <iostream>

using namespace std;

int main() {

ifstream inputFile("Number.txt");

ofstream outputFile("Num.txt");

if (!inputFile.is\_open() || !outputFile.is\_open()) {

cout << "Error opening file" << endl;

return 1;

}

double prev, current, next;

inputFile >> prev;

outputFile << prev << endl;

while (inputFile >> current) {

inputFile >> next;

outputFile << (prev + current + next) / 2 << endl;

prev = current;

}

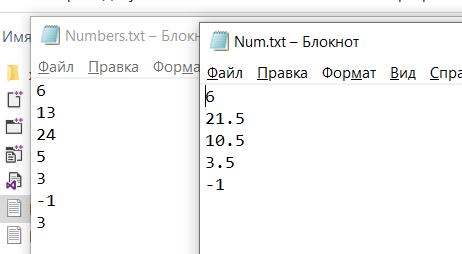
outputFile << current << endl;

inputFile.close();

outputFile.close();

return 0;

}



Дан файл целых чисел, содержащий более 5 элементов. Уменьшить его размер до 5 элементов, удалив из файла необходимое количество конечных элементов.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

int num, count = 0;

int arr[100];

ifstream fin;

fin.open("Numbers.txt");

while (fin >> num) {

arr[count] = num;

count++;

}

fin.close();

ofstream fout;

fout.open("Num.txt");

int n = count > 5 ? 5 : count;

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

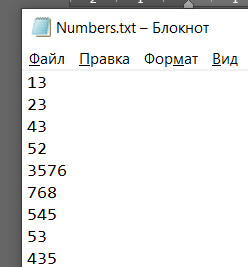
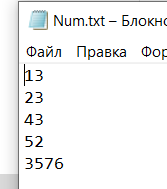
fout << arr[i] << endl;

}

fout.close();

return 0;

}

Дан файл целых чисел. Удалить из него все элементы с четными номерами.

#include <iostream>

#include <fstream>

#include <vector>

int main() {

std::ifstream input("Numbers.txt");

std::ofstream output("Num.txt");

std::vector<int> numbers;

int x;

while (input >> x) {

numbers.push\_back(x);

}

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

if (i % 2 != 0) {

output << numbers[i] << ' ';

}

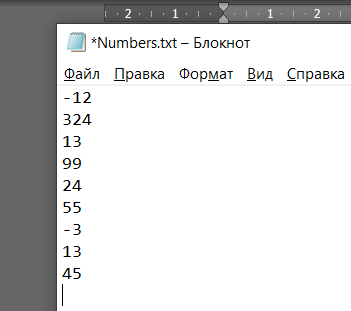
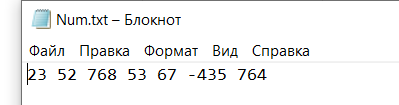
}

input.close();

output.close();

return 0;

}

Дан файл целых чисел. Удвоить его размер, записав в конец файла все его исходные элементы(в том же порядке).

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main()

{

vector<int> numbers;

int x;

ifstream inputFile("Numbers.txt");

if (!inputFile)

{

cout << "File not found" << endl;

return 1;

}

while (inputFile >> x)

{

numbers.push\_back(x);

}

inputFile.close();

ofstream outputFile("Num.txt", ios::app);

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

{

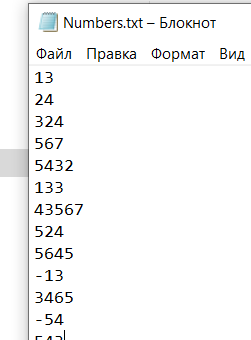
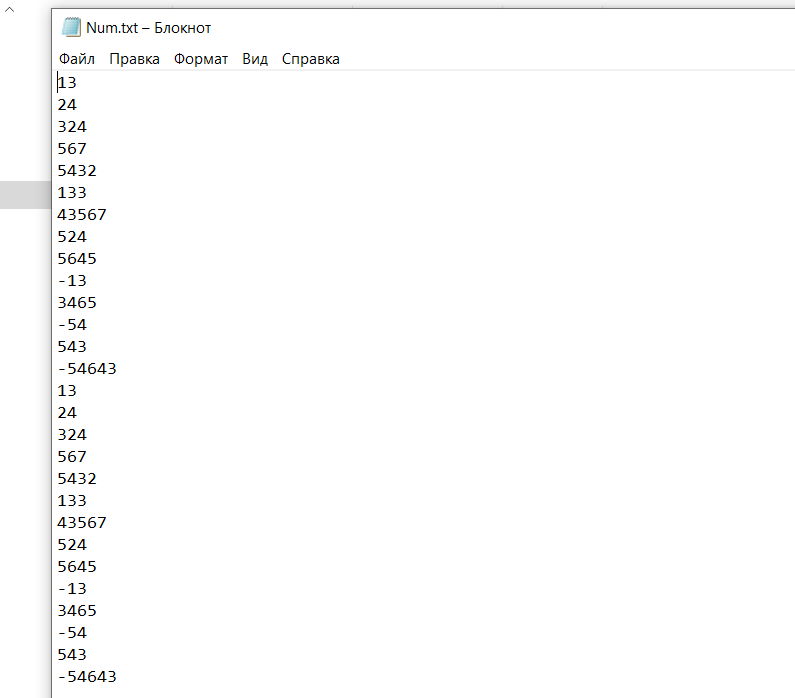
outputFile << numbers[i] << endl;

}

outputFile.close();

return 0;

}

Дан файл целых чисел. Продублировать в нем все элементы с нечетными номерами.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

vector<int> numbers;

int x;

ifstream input("Number.txt");

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

cerr << "Error: Unable to open the file." << endl;

return 1;

}

while (input >> x) {

numbers.push\_back(x);

}

input.close();

ofstream output("Num.txt");

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

cerr << "Error: Unable to open the file." << endl;

return 1;

}

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

output << numbers[i] << endl;

if (i % 2 != 0) {

output << numbers[i] << endl;

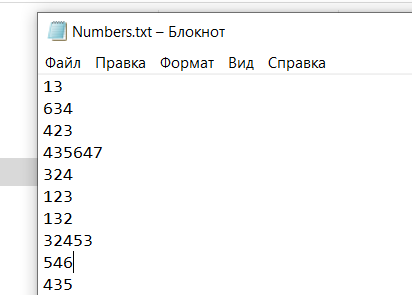
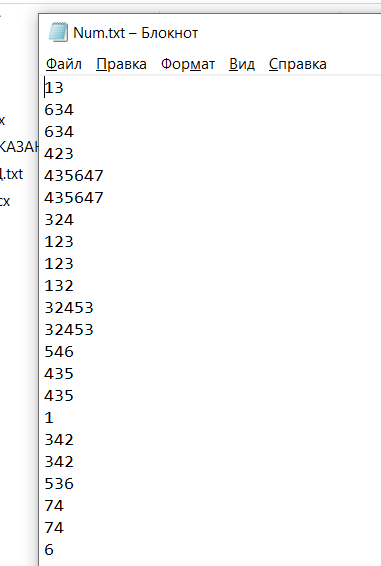
}

}

output.close();

return 0;

}

Дан файл целых чисел. Продублировать в нем все числа, принадлежащие диапазону 5 – 10.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

ifstream input\_file("Numbers.txt");

ofstream output\_file("Num.txt");

vector<int> numbers;

int number;

while (input\_file >> number) {

numbers.push\_back(number);

}

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

if (numbers[i] >= 5 && numbers[i] <= 10) {

output\_file << numbers[i] << endl;

output\_file << numbers[i] << endl;

}

else {

output\_file << numbers[i] << endl;

}

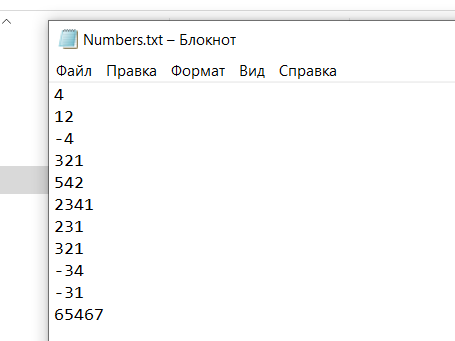
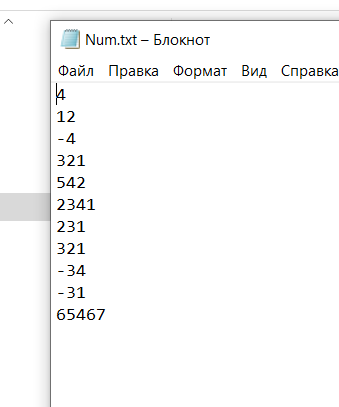
}

input\_file.close();

output\_file.close();

return 0;

}

Дан файл целых чисел. Заменить в нем каждый элемент с четным номером на два нуля.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ifstream inputFile("Numbers.txt");

ofstream outputFile("Num.txt");

int number;

int count = 0;

while (inputFile >> number) {

count++;

if (count % 2 == 0) {

outputFile << "0 0" << endl;

}

else {

outputFile << number << endl;

}

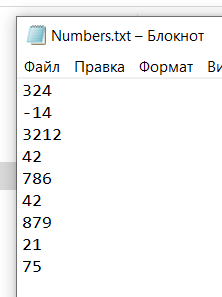
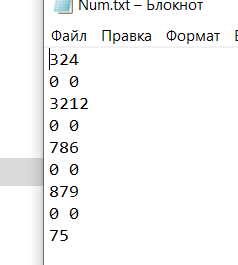
}

inputFile.close();

outputFile.close();

return 0;

}

Дан файл целых чисел. Заменить в нем каждое положительное число на три нуля.

#include <fstream>

#include <iostream>

int main() {

std::ifstream inputFile("Numbers.txt");

std::ofstream outputFile("Num.txt");

int number;

while (inputFile >> number) {

if (number > 0) {

outputFile << "0 0 0 ";

}

else {

outputFile << number << " ";

}

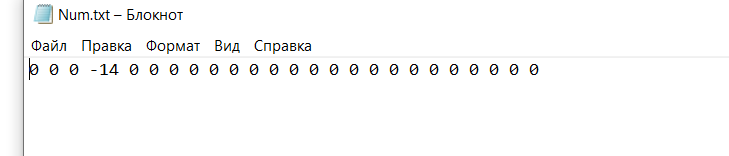
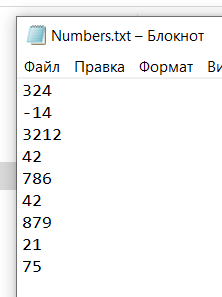
}

inputFile.close();

outputFile.close();

return 0;

}



**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №11

по МДК 01.01 Системное программирование

на тему

Нетипизированные файлы. Работа с нетипизированными файлами.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <fstream>

using namespace std;

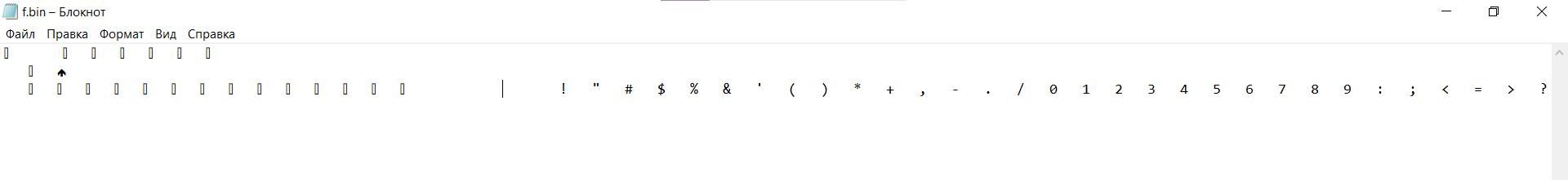
iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

fstream – заголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

В ходе работы нет смысла скриншотить выходные файлы, так как там бинарная информация:



Составить программу, которая создает нетипизированный файл из 100 чисел и выводит на экран k-ый элемент.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

ofstream file("f", ios::out | ios::binary);

for (int i = 1; i <= 100; i++) {

file.write(reinterpret\_cast<const char\*>(&i), sizeof(i));

}

file.close();

int k;

cout << "Введите к:\t";

cin >> k;

ifstream infile("f", ios::in | ios::binary);

infile.seekg((k - 1) \* sizeof(int));

int num;

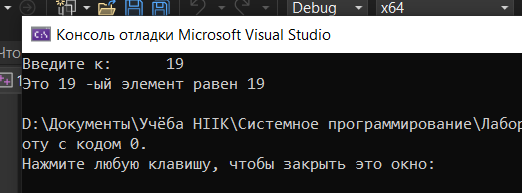
infile.read(reinterpret\_cast<char\*>(&num), sizeof(num));

infile.close();

cout << "Это " << k << " -ый элемент равен " << num << endl;

return 0;

}



Составить программу, которая создает копию элементов нетипизированного файла f и помещает в файл g.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

ifstream infile("f", ios::in | ios::binary);

if (!infile) {

cerr << "Не могу найти файл f" << endl;

return 1;

}

ofstream outfile("g", ios::out | ios::binary);

if (!outfile) {

cerr << "Не могу открыть файл g" << endl;

return 1;

}

char buffer[1024];

while (infile.read(buffer, sizeof(buffer))) {

outfile.write(buffer, infile.gcount());

}

outfile.write(buffer, infile.gcount());

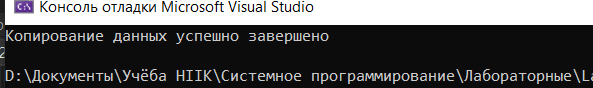
infile.close();

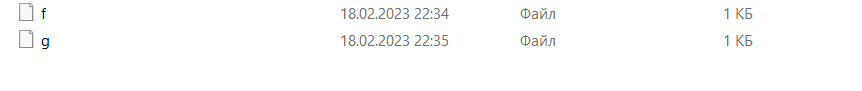
outfile.close();

cout << "Копирование данных успешно завершено" << endl;

return 0;

}





Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет среднее арифметическое элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int sum = 0, count = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

sum += num;

count++;

}

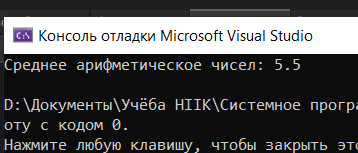
infile.close();

double average = static\_cast<double>(sum) / count;

cout << "Среднее арифметическое чисел: " << average << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет сумму элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int sum = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

sum += num;

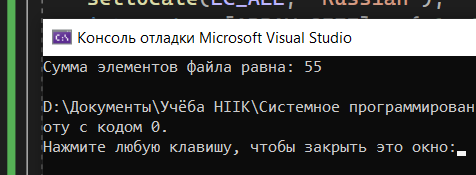
}

infile.close();

cout << "Сумма элементов файла равна: " << sum << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет произведение элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int product = 1;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

product \*= num;

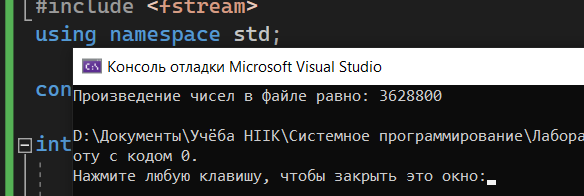
}

infile.close();

cout << "Произведение чисел в файле равно: " << product << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет количество чётных элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int count = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

if (num % 2 == 0) {

count++;

}

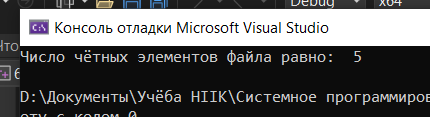
}

infile.close();

cout << "Число чётных элементов файла равно: " << count << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет количество элементов меньше числа К файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int count = 0;

int k;

cout << "Введите k:\t";

cin >> k;

cout << endl;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

if (num < k) {

count++;

}

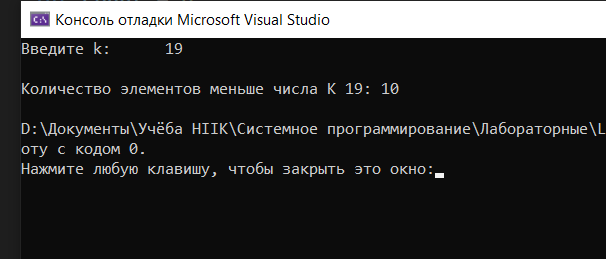
}

infile.close();

cout << "Количество элементов меньше числа K " << k << ": " << count << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет среднее арифметическое элементов файла и сумму элементов меньше среднего арифметического.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int sum = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

sum += num;

}

infile.close();

double average = static\_cast<double>(sum) / ARRAY\_SIZE;

int sum\_less\_than\_average = 0;

infile.open("numbers", ios::in | ios::binary);

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

if (num < average) {

sum\_less\_than\_average += num;

}

}

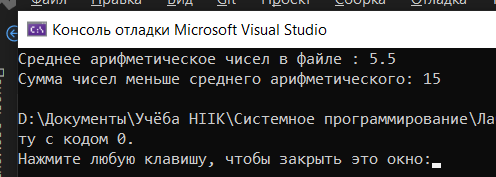
infile.close();

cout << "Среднее арифметическое чисел в файле : " << average << endl;

cout << "Сумма чисел меньше среднего арифметического: " << sum\_less\_than\_average << endl;

return 0;

}



**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №12

по МДК 01.01 Системное программирование

на тему

Описание процедур и функций.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

using namespace std;

iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

#include <iostream>

using namespace std;

float distance(float x1, float y1, float x2, float y2)

{

return sqrt((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));

}

bool multiple(int x, int y)

{

if (x % y == 0)

return true;

else

return false;

}

void print(int x)

{

cout << "x=" << x << endl;

}

void f1();

void f2();

void f3();

int main() {

setlocale(LC\_ALL, "Russian");

f1();

cout << endl;

f2();

cout << endl;

f3();

cout << endl;

return EXIT\_SUCCESS;

}

void f1()

{

float a, b, c, d, rez;

cout << "Введите координаты точек: " << endl;

cin >> a >> b >> c >> d;

rez = distance(a, b, c, d);

cout << "Результат: " << rez << endl;

}

void f2()

{

int a, b; bool rez;

cout << "Введите числа: " << endl;

cin >> a >> b;

rez = multiple(a, b);

if (rez)

cout << "Первое число кратно второму" << endl;

else

cout << "Первое число некратно второму" << endl;

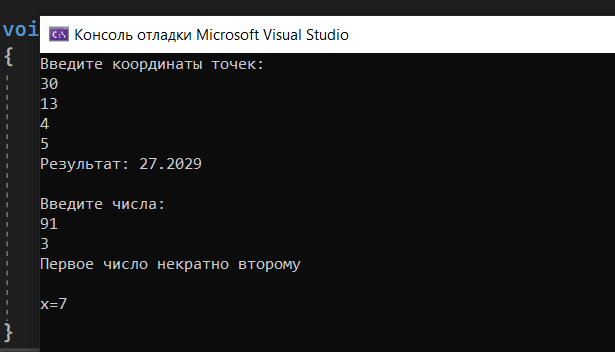
}

void f3()

{

print(7);

}



**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №13

по МДК 01.01 Системное программирование

на тему

Организация функций.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <cmath>

#include <iomanip>;

using namespace std;

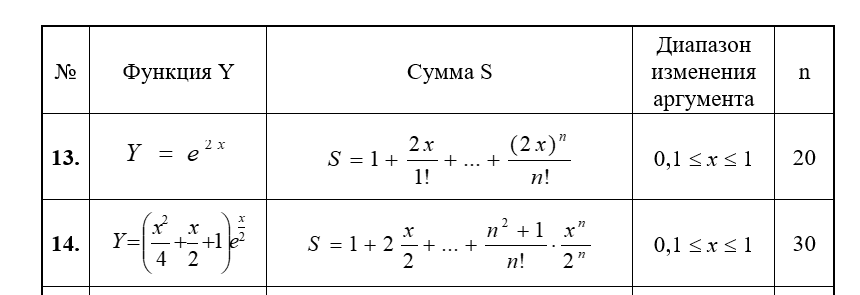
iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

cmath — заголовочный файл стандартной библиотеки разработанный для выполнения простых математических операций.

iomanip - заголовочный файл стандартной библиотеки, в котором описаны манипуляторы при работе с потоковыми операциями.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

****

**Вариант 14**

#include <iostream>

#include <cmath>

#include <iomanip>;

using namespace std;

double factorial(int n) {

double result = 1.0;

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

result \*= i;

}

return result;

}

double nonRecursiveSum(double x, int n) {

double s = 1.0;

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

double term = (pow(i, 2) + 1) / factorial(i) \* pow(x / 2.0, i);

s += term;

}

return s;

}

double recursiveSum(double x, int n, double s = 1.0, int i = 1) {

if (i > n) {

return s;

}

double term = (pow(i, 2) + 1) / factorial(i) \* pow(x / 2.0, i);

s += term;

return recursiveSum(x, n, s, i + 1);

}

int main() {

setlocale(LC\_ALL, "RU");

double xk = 0.1; // Начальное значение

double xh = 1.0; // Конечное значение

double h = (xh - xk) / 10.0; // Шаг

int n = 30; // Число элементов для суммы

const double MATH\_E = 2.71828182846; // Константа E = 2,71

cout << setw(5) << "x" << setw(15) << "y" << setw(20) << "s" << endl;

cout << "------------------------------------------------------------" << endl;

double y, s;

for (double x = xk; x <= xh; x += h) {

// Формула Y = (x^2 / 4 + x / 2 + 1) \* e^(x/2)

y = (pow(x, 2) / 4.0 + x / 2.0 + 1.0) \* pow(MATH\_E,x / 2);

// Вычисление суммы через цикл for

s = nonRecursiveSum(x, n);

cout << "| x = " << x << "| , | y = " << y << "| , | s (через цикл) = " << s << "|" << endl;

// Вычисление суммы через рекурсию

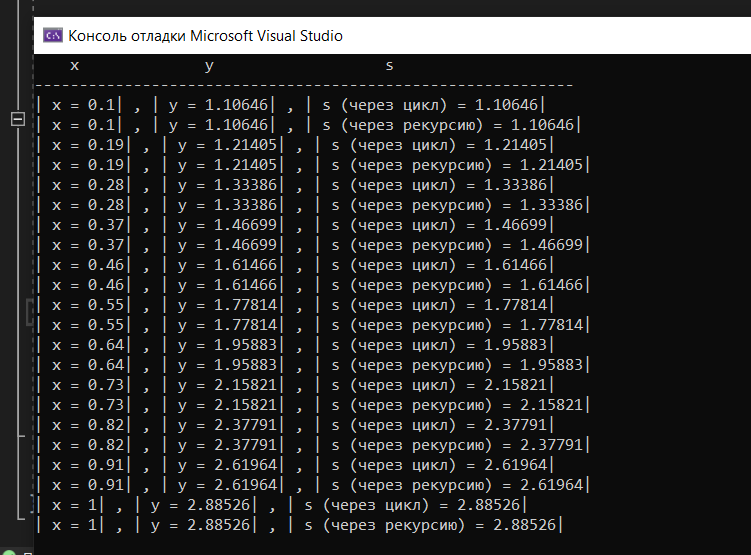
s = recursiveSum(x, n);

cout << "| x = " << x << "| , | y = " << y << "| , | s (через рекурсию) = " << s << "|" << endl;

}

return 0;

}

****

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №14

по МДК 01.01 Системное программирование

на тему

Применение рекурсивных функций.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <fstream>

#include <cmath>

using namespace std;

iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

fstream – заголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл.

cmath — заголовочный файл стандартной библиотеки разработанный для выполнения простых математических операций.

iomanip - заголовочный файл стандартной библиотеки, в котором описаны манипуляторы при работе с потоковыми операциями.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

Для удобства проверки, задания выполненны в одном файле .cpp.

1. Ввести последовательность чисел (окончание ввода – 0) и вывести их в обратной последовательности. Входные данные взять из текстового файла.
2. Используя cout <<x лишь при x=0..9, написать рекурсивную программу печати десятичной записи целого положительного числа n.
3. Напишите рекурсивную функцию, которая возвращает среднее из n элементов массива чисел.
4. Найти первые N чисел Фибоначчи двумя способами: с помощью рекурсии и с помощью итерации. Сравнить эффективность алгоритмов.
5. Написать функцию сложения двух чисел, используя только прибавление единицы.
6. Написать функцию умножения двух чисел, используя только операцию сложения.
7. Вычислить сумму элементов одномерного массива.
8. Найти НОД (наибольший общий делитель) двух натуральных чисел.
9. Проверить, является ли фрагмент строки с *i*-го по *j*-й символ палиндромом.
10. Вычислить произведение элементов одномерного массива.
11. Написать функцию сортировки массива методом простого выбора.
12. Подсчитать количество цифр в заданном числе.
13. Написать функцию, проверяющую правильность имени в языке Pascal.
14. Написать функцию определения, является ли заданное натуральное число простым.

#include <iostream>

#include <fstream>

#include <vector>

#include <math.h>

#include <cmath>

using namespace std;

// Задание 1 : Вывести в обратном порядке из файла

void reverseOrder(ifstream& infile) {

int num;

infile >> num;

if (num != 0) {

reverseOrder(infile);

cout << num << " ";

}

}

// Задание 2 : Числовое представление

void decimalRepresentation(int n) {

if (n < 10) {

cout << n;

}

else {

decimalRepresentation(n / 10);

cout << n % 10;

}

}

// Задание 3 : Среднее по массиву

double averageArray(int arr[], int n) {

if (n == 1) {

return arr[0];

}

else {

return ((arr[n - 1] + averageArray(arr, n - 1) \* (n - 1)) / n);

}

}

// Задание 4 : Числа Фиббоначи

void fibRecursive(int n, int& prev, int& curr) {

if (n == 0) {

curr = 0;

prev = 1;

}

else {

int tempPrev, tempCurr;

fibRecursive(n - 1, tempPrev, tempCurr);

curr = tempPrev + tempCurr;

prev = tempCurr;

}

}

void fibIterative(int n) {

int prev = 0, curr = 1;

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

int temp = prev;

prev = curr;

curr = temp + curr;

}

cout << prev << " ";

}

// Задание 5 : Сложить два числа

int add(int a, int b) {

if (b == 0) {

return a;

}

else {

return add(a ^ b, (a & b) << 1);

}

}

// Задание 6 : Умножить два числа

int multiply(int a, int b) {

if (b == 0) {

return 0;

}

else if (b % 2 == 0) {

return multiply(a + a, b / 2);

}

else {

return add(a, multiply(a + a, (b - 1) / 2));

}

}

// Задание 7 : Сумма элементов массива

int sumArray(int arr[], int n) {

if (n == 0) {

return 0;

}

else {

return arr[n - 1] + sumArray(arr, n - 1);

}

}

// Задание 8 : НОД

int gcd(int a, int b) {

if (b == 0) {

return a;

}

else {

return gcd(b, a % b);

}

}

// Задание 12 : Палиндром

bool isPalindrome(string str, int i, int j) {

if (i >= j) {

return true;

}

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

return false;

}

return isPalindrome(str, i + 1, j - 1);

}

// Задание 13 : Произведение элементов

int arrayProduct(int arr[], int n) {

if (n == 0) {

return 1;

}

return arr[n - 1] \* arrayProduct(arr, n - 1);

}

// Задание 14 : Сортировка массива

void selectionSort(int arr[], int n) {

if (n <= 1) {

return;

}

int maxIdx = 0;

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

if (arr[i] > arr[maxIdx]) {

maxIdx = i;

}

}

swap(arr[maxIdx], arr[n - 1]);

selectionSort(arr, n - 1);

}

// Задание 15 : Количество цифр в заданном числе

int countDigits(int n) {

if (n == 0) {

return 0;

}

return 1 + countDigits(n / 10);

}

// Задание 16 : Правильность имён на Паскале

bool isValidPascalName(string name) {

if (name.length() == 0) {

return false;

}

if (!isupper(name[0])) {

return false;

}

for (int i = 1; i < name.length(); i++) {

if (!islower(name[i]) && !isdigit(name[i])) {

return false;

}

}

return true;

}

// Задание 18 : Простое число

bool isPrime(int n, int i = 2) {

if (n <= 2) {

return (n == 2) ? true : false;

}

if (n % i == 0) {

return false;

}

if (i \* i > n) {

return true;

}

return isPrime(n, i + 1);

}

// Задание 19 : Цепочка построения домино

void dominoChain(int dominoes[], int n, int last, bool used[], string& chain) {

if (chain.length() == n \* 2) {

cout << chain << endl;

return;

}

for (int i = 0; i < n \* 2; i++) {

if (!used[i]) {

int a = dominoes[i / 2];

int b = dominoes[i / 2 + (i % 2 ? -1 : 1)];

if (a == last) {

used[i] = true;

chain.push\_back('0' + a);

chain.push\_back('0' + b);

dominoChain(dominoes, n, b, used, chain);

chain.pop\_back();

chain.pop\_back();

used[i] = false;

}

else if (b == last) {

used[i] = true;

chain.push\_back('0' + b);

chain.push\_back('0' + a);

dominoChain(dominoes, n, a, used, chain);

chain.pop\_back();

chain.pop\_back();

used[i] = false;

}

}

}

}

int main() {

setlocale(LC\_ALL, "RU");

ifstream infile("input.txt");

// Задание 1 : Вывести в обратном порядке из файла

cout << "Задание 1 : Вывести в обратном порядке из файла" << endl;

reverseOrder(infile);

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 2 : Числовое представление

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

int n;

cout << "Введите положительное число: ";

cin >> n;

cout << "Десятичная запись: ";

decimalRepresentation(n);

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 3 : Среднее по массиву

cout << "Задание 3 : Среднее по массиву" << endl;

int Array[] = { 1, 2, 3, 4, 5 };

int n2 = sizeof(Array) / sizeof(Array[0]);

cout << "Среднее по массиву : " << averageArray(Array, n2) << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 4 : Числа Фиббоначи

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

cout << "Используя рекурсию: ";

int prev = 0, curr = 0;

for (int i = 0; i < 10; i++) {

fibRecursive(i, prev, curr);

cout << curr << " ";

}

cout << endl;

cout << "Используя итерации: ";

fibIterative(10);

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 5 : Сложить два числа

cout << "Задание 5 : Сложить два числа" << endl;

int a, b;

cout << "Введите два числа для сложения: ";

cin >> a >> b;

cout << "Сумма: " << add(a, b) << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 6 : Умножить два числа

cout << "Задание 6 : Умножить два числа" << endl;

cout << "Введите два числа для умножения: ";

cin >> a >> b;

cout << "Произведение: " << multiply(a, b) << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 7 : Сумма элементов массива

cout << "Задание 7 : Сумма элементов массива" << endl;

int MyArray[] = { 10, 20, 30, 40, 50 };

int n3 = sizeof(MyArray) / sizeof(MyArray[0]);

cout << "Сумма элементов: " << sumArray(MyArray, n3) << endl;

// Задание 8 : НОД

cout << "Задание 8 : НОД" << endl;

int a2, b2;

cout << "Введите два положительных числа: ";

cin >> a2 >> b2;

cout << "НОД: " << gcd(a2, b2) << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 12 : Палиндром

cout << "Задание 12 : Палиндром" << endl;

string str = "racecar";

int i = 0;

int j = str.length() - 1;

if (isPalindrome(str, i, j)) {

cout << str.substr(i, j - i + 1) << " палиндром " << endl;

}

else {

cout << str.substr(i, j - i + 1) << " не палиндром " << endl;

}

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 13 : Произведение элементов массива

cout << "Задание 13 : Произведение элементов массива" << endl;

int ByMyG[] = { 1, 2, 3, 4, 5 };

n = sizeof(ByMyG) / sizeof(ByMyG[0]);

int product = arrayProduct(ByMyG, n);

cout << "Произведение элементов массива: " << product << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 14 : Сортировка массива

cout << "Задание 14 : Сортировка массива" << endl;

int LoP[] = { 5, 3, 8, 6, 7, 2 };

int len = sizeof(LoP) / sizeof(LoP[0]);

selectionSort(LoP, len);

cout << "Отсортированный массив ";

for (int i = 0; i < n2; i++) {

cout << LoP[i] << " ";

}

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 15 : Количество цифр в заданном числе

cout << "Задание 14 : Количество цифр в заданном числе" << endl;

int num = 12345;

int numDigits = countDigits(num);

cout << "Количество цифр числа " << num << " равно " << numDigits << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 16 : Правильность имён на Паскале

cout << "Задание 16 : Правильность имён на Паскале" << endl;

string name = "ValidName1";

if (isValidPascalName(name)) {

cout << name << " возможное имя на языке Паскаль" << endl;

}

else {

cout << name << " так писать на Паскале нельзя! " << endl;

}

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

// Задание 18 : Простое число

cout << "Задание 18 : Простое число" << endl;

int num2 = 17;

if (isPrime(num2)) {

cout << num2 << " простое число " << endl;

}

else {

cout << num2 << " непростое число " << endl;

}

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

return 0;

}

