**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++)

{

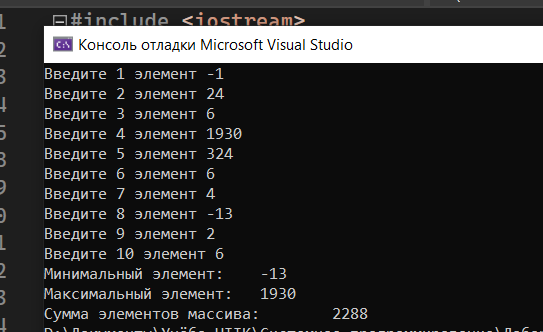
sumResult += myArr[i];

}

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;

}

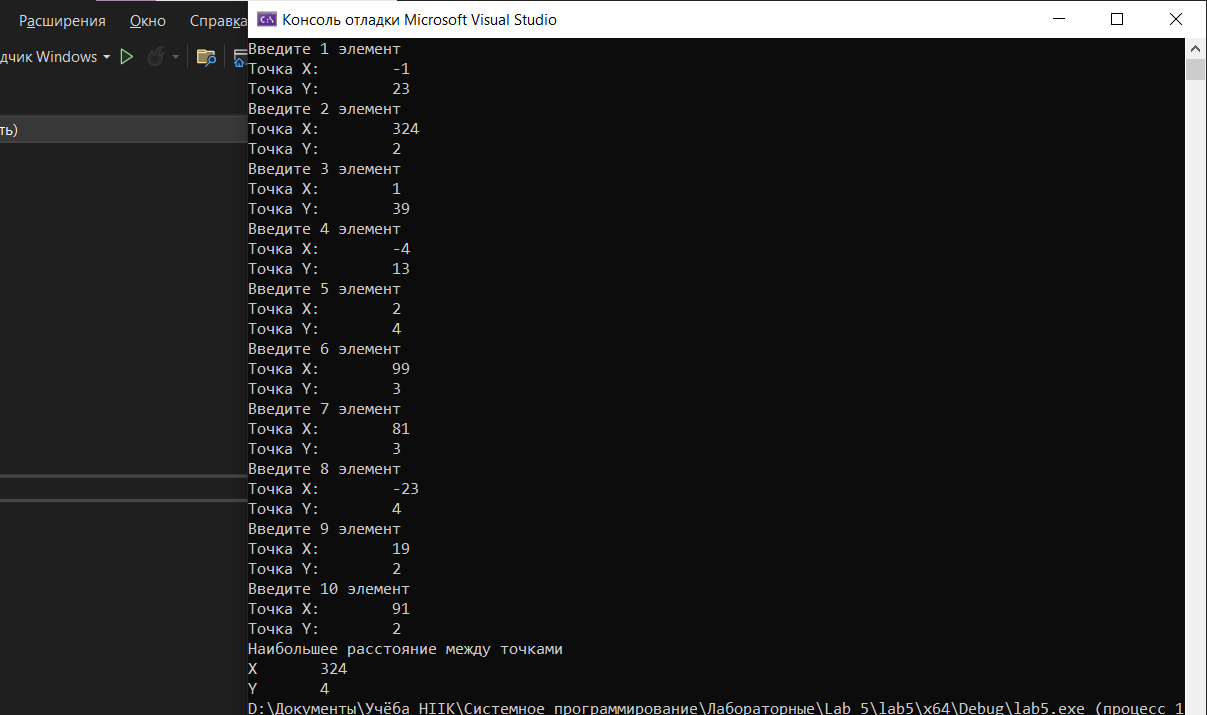
}

}

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++)

{

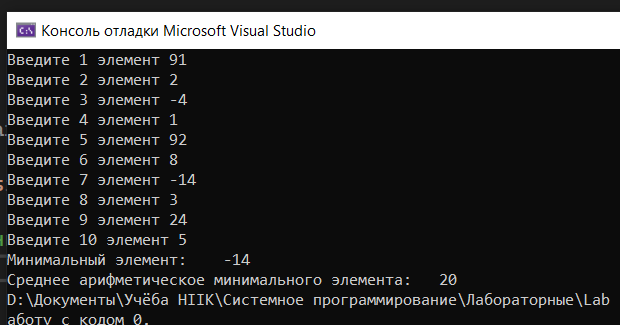
sumResult += A[i];

}

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)

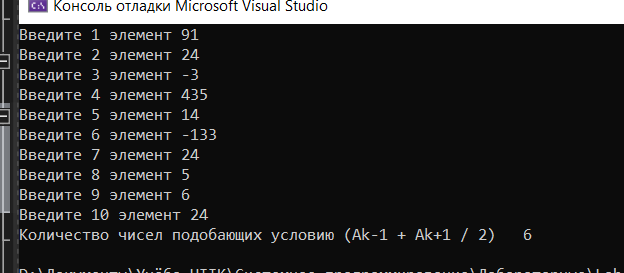
{

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))

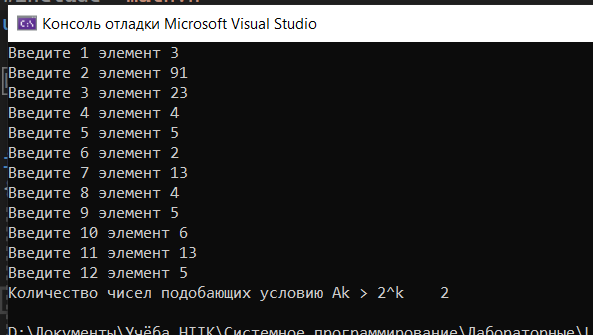
{

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)

{

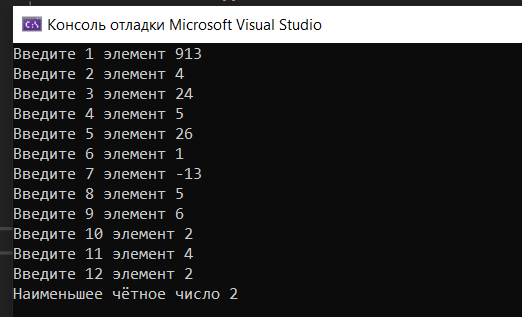
min = A[i];

}

}

}

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)

{

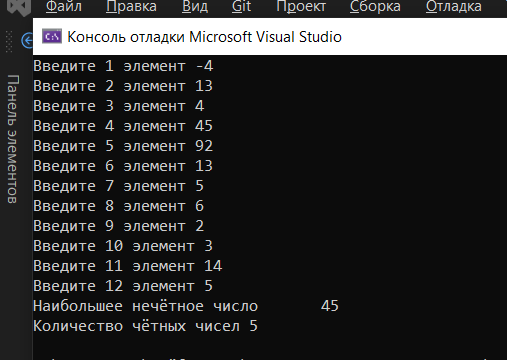
counter++;

}

}

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;

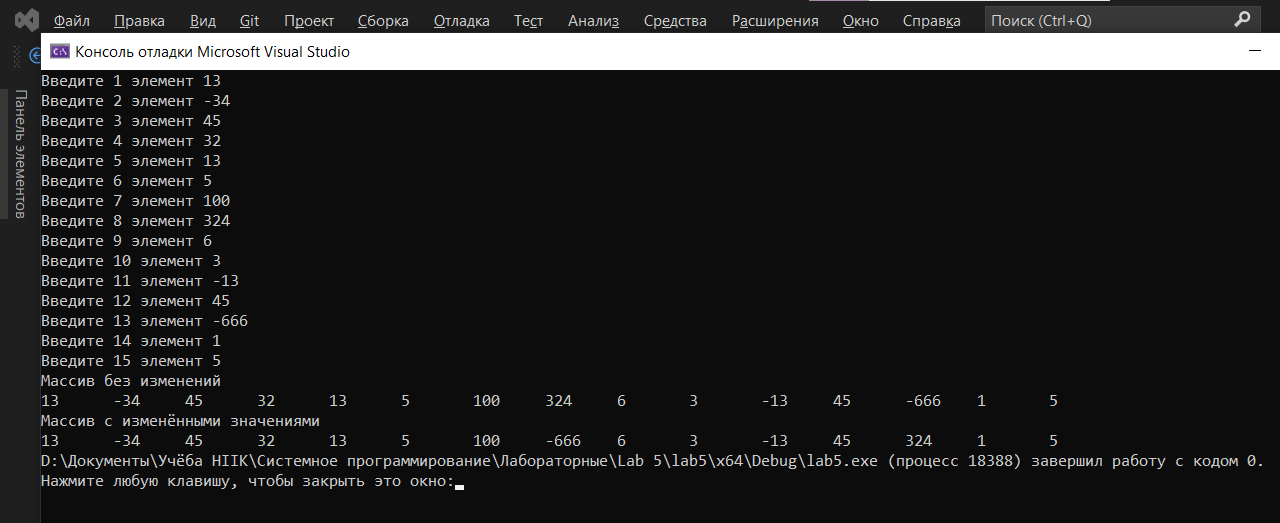
cout << "Массив с изменёнными значениями\n";

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;

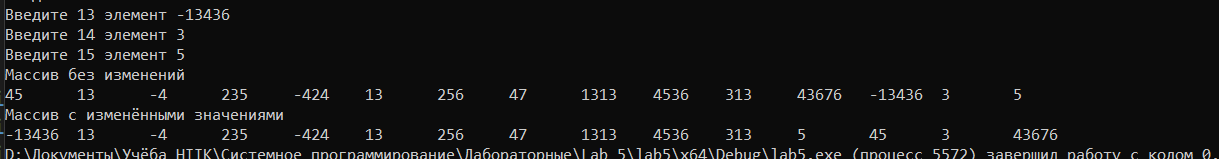
cout << "Массив с изменёнными значениями\n";

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;

}

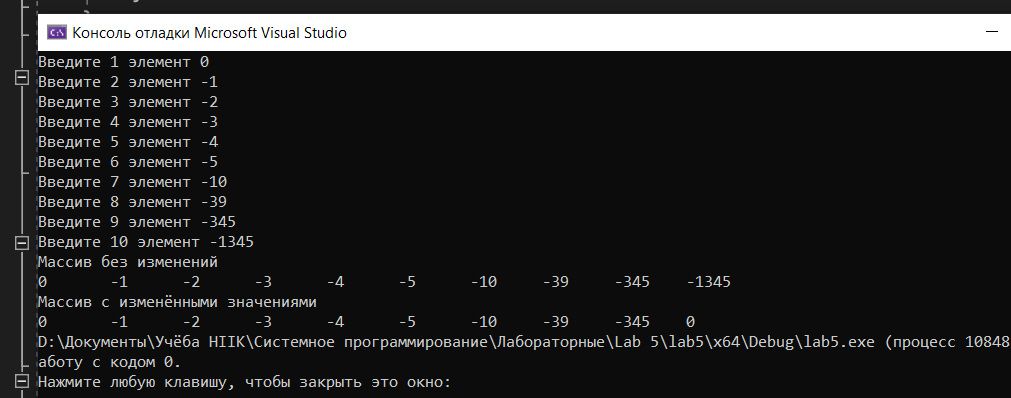
cout << "Массив с изменёнными значениями\n";

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";

}

