**Вычислительные задачи.**

**Задача 5.40**

Условие: Даны натуральное число n и вещественные a1,a2,…,an.Определить сумму всех вещественных чисел.

Решение:

Словесное описание алгоритма:

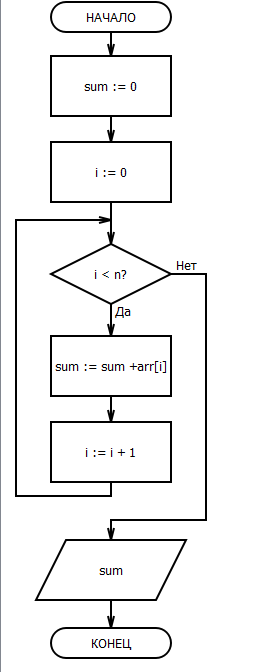
1.Инициализируем переменную n и статический массив вещественных чисел размером n.

2. Инициализируем переменную sum нулем, она нужна для хранения суммы.

3. В цикле от 0 до n (не включая n) прибавляем к sum каждый элемент массива.

4. Выводим результат на экран.

Блок-схема алгоритма:



Программная реализация алгоритма на языке C++:

#include<iostream>

using namespace std;

int main()

{

const int n = 4;

float arr[] = {1.43f,3.234f,534.23f,99.01f};

float sum = 0;

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

{

sum += arr[i];

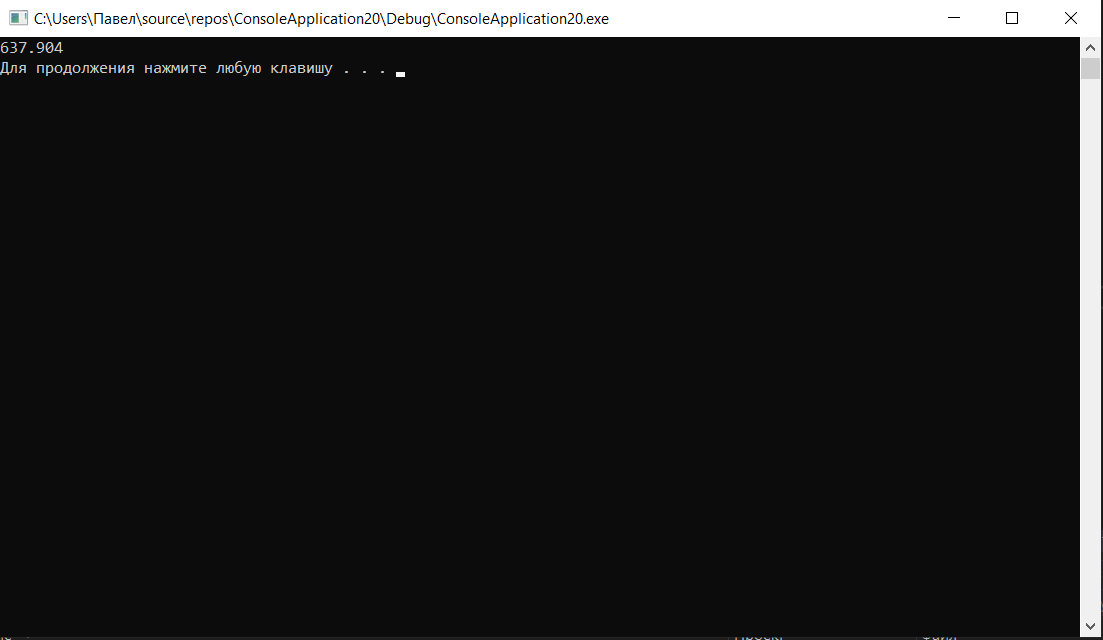
}

cout<<sum;

return 0;

}

Листинг программы:



**Задача 7.48**

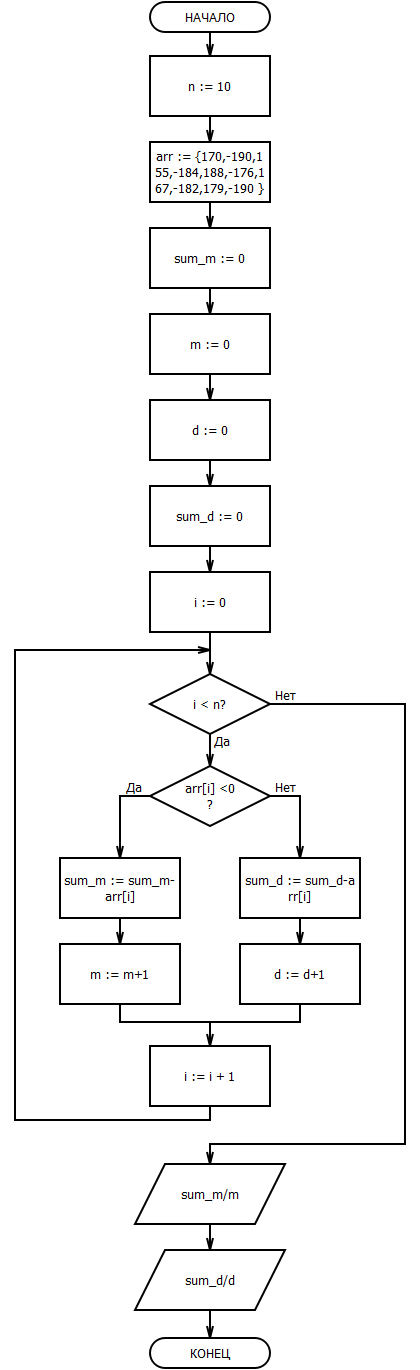
Условие: . Известен рост каждого ученика класса. Рост мальчиков условно задан отрицательными числами. Определить средний рост мальчиков и средний рост девочек.

Решение:

Словесное описание алгоритма:

1. Инициализируем статический целочисленный массив, который хранит рост всех учеников и переменную n, значение которой равно числу учеников.
2. Инициализируем нулями переменные, которые будут хранить сумму роста мальчиков, их количество, сумму роста девочек, их количество.
3. В цикле от 0 до n, проверяем, если рост меньше нуля, то умножаем его на -1 и суммируем к росту мальчиков и к их количеству добавляем единицу.
4. Если рост больше 0, то добавляем это значение к росту девочек, а их количество увеличиваем на единицу.
5. Выводим на экран средний рост мальчиков (отношение суммы роста к количеству мальчиков).
6. Выводим средний рост девочек (по аналогии с мальчиками).

Блок-схема алгоритма:



Программная реализация алгоритма:

#include<iostream>

using namespace std;

int main()

{

const int n = 10;

int arr[10] = { 170,-190,155,-184,188,-176,167,-182,179,-190 };

double sum\_m = 0;

int m = 0;

double sum\_d = 0;

int d = 0;

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

if (arr[i] < 0)

{

sum\_m += (-1) \* arr[i];

m++;

}

else

{

sum\_d += arr[i];

d++;

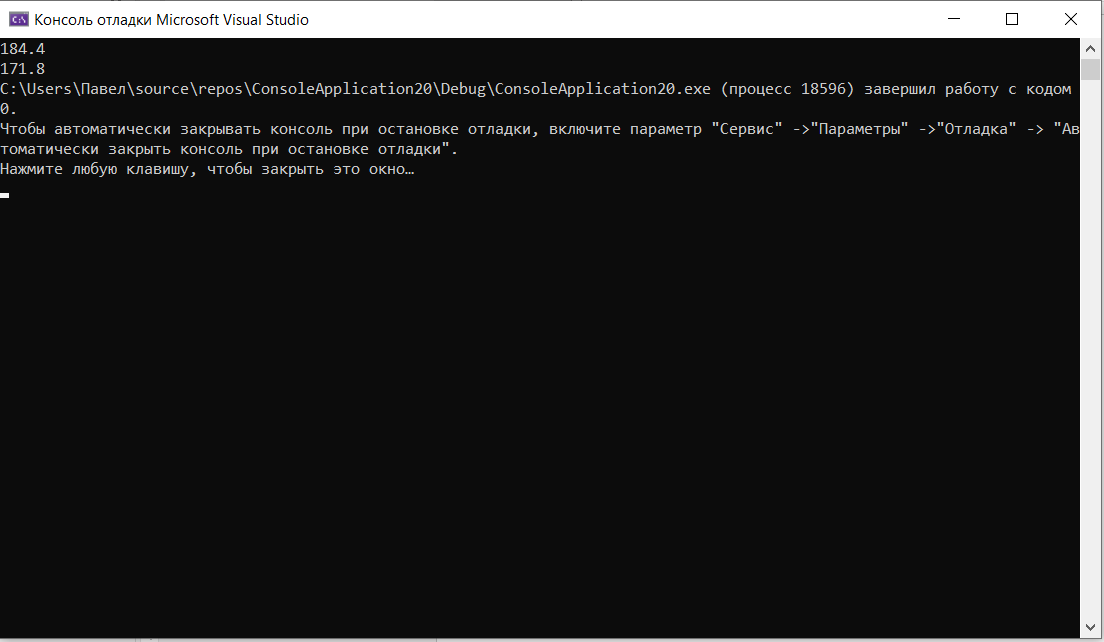
}

cout << sum\_m / m << endl << sum\_d / d;

return 0;

}

Листинг программы:



**Задача 9.44**

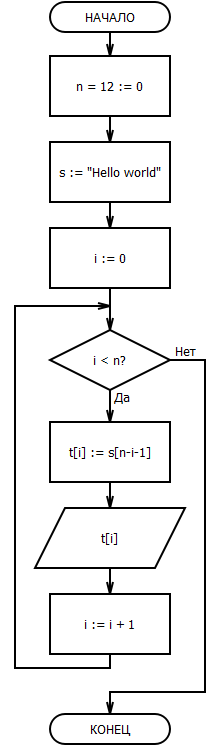
Условие: Дано слово s. Получить слово t, получаемое путем прочтения слова s начиная с его конца.

Решение:

Словестное описание алгоритма:

1. Инициализируем массив символов, из которых состоит слово и переменную, значение которой равно длине слова.
2. Объявляем массив символов для хранения нового слова размером исходного массива.
3. В цикле, пробегаясь по всем элементам исходного массива, записываем их в новый с конца.
4. Выводим новое слово на экран.

Блок-схема алгоритма:



Программная реализация алгоритма:

#include<iostream>

using namespace std;

int main()

{

const int n = 12;

char s[n] = "Hello world";

char t[n];

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

{

t[i] = s[n - i - 1];

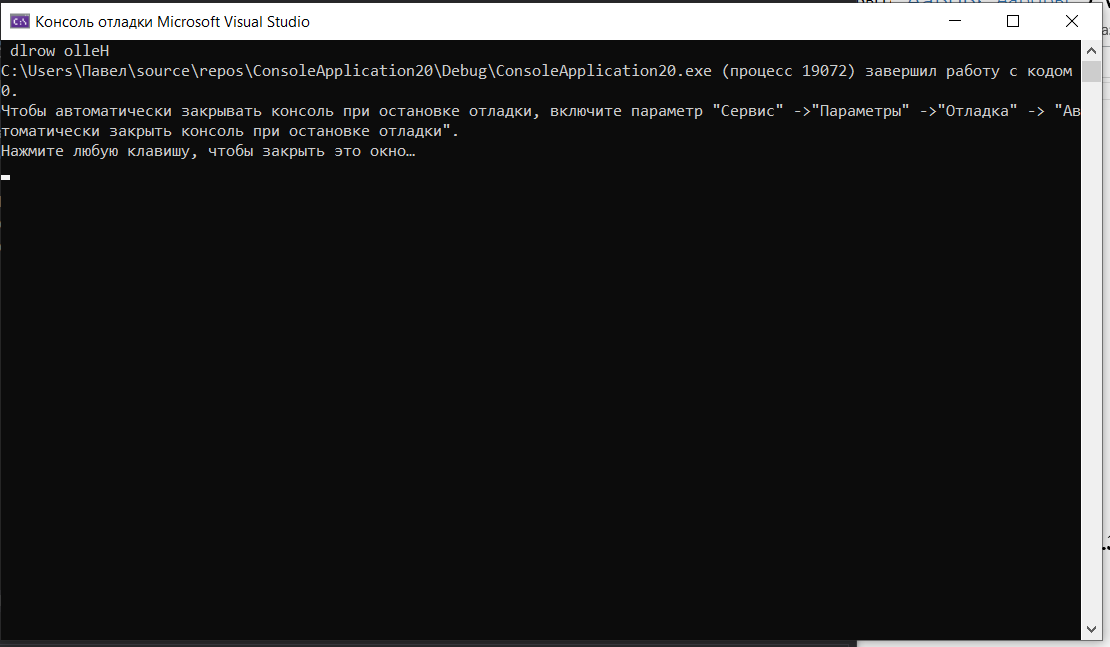
cout << t[i];

}

return 0;

}

Листинг программы:



**Задача 10.32**

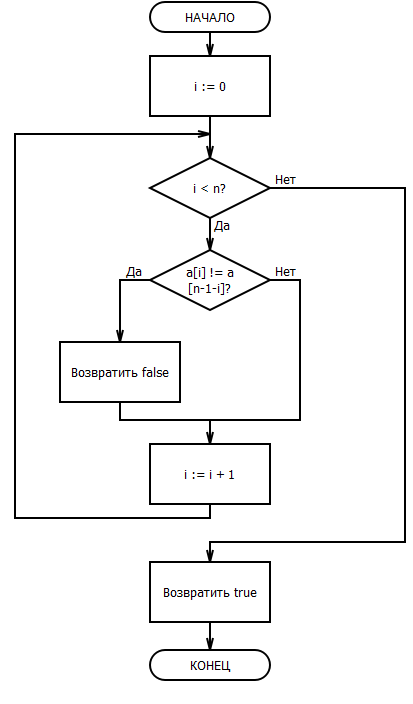
Условие: Даны три слова. Выяснить, является ли хоть одно из них палиндромом ("перевертышем"), т. е. таким, которое читается одинаково слева направо и справа налево. (Определить функцию, позволяющую распознавать слова палиндромы.)

Решение:

Словесное описание алгоритма:

1. Определим функцию, принимающую в качестве аргументов константный указатель на массив символов и его размер, и возвращающую результат сравнения.
2. Очевидно, для того чтобы слово являлось палиндромом, при проходе массива в обе стороны соответствующие символы будут равны.
3. При первом нарушении равенства прерываем функцию и возвращаем false;
4. Если условие пройдено успешно всеми элементами массива, то возвращается true;

Блок-схема алгоритма:



Программная реализация алгоритма:

#include<iostream>

using namespace std;

bool Palindrome(const char\* a,int n)

{

for (int i = 0; i < n/2; i++) if (a[i] != a[n - 1 - i]) return false;

return true;

}

int main()

{

char a[4] = "LoL";

char b[9] = "Darkness";

char c[12] = "aaaabbbaaaa";

cout << Palindrome(a, 3) << endl << Palindrome(b, 8) << endl << Palindrome(c, 11);

return 0;

}

Листинг программы:

