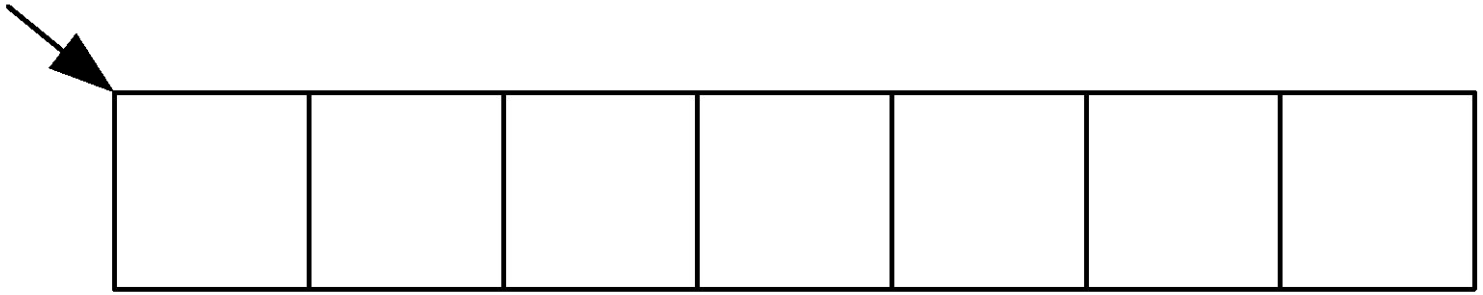
Одномерные статические массивы в C++

Часть 1. Синтаксис и простейшие задачи

Представление массива



Массив – это совокупность элементов одного типа данных, расположенных в памяти друг за другом и имеющих одно имя.

Статическим называют массив, память под ячейки которого выделяется статически, а число элементов массива не меняется в ходе работы программы.

Статическое выделение памяти выполняется для статических и глобальных объектов, осуществляется только один раз (при запуске программы), состояние объектов сохраняется на протяжении работы всей программы.

Встроенные массивы

тип имя_массива [размер_массива];

В качестве размера статического массива обязательно должна использоваться целочисленная константа.

Например целочисленные литералы:

```
int mas[7];
```

или именованные целочисленные константы:

```
const int n = 5;
float a[n];
```

Разные способы задания размерности массива

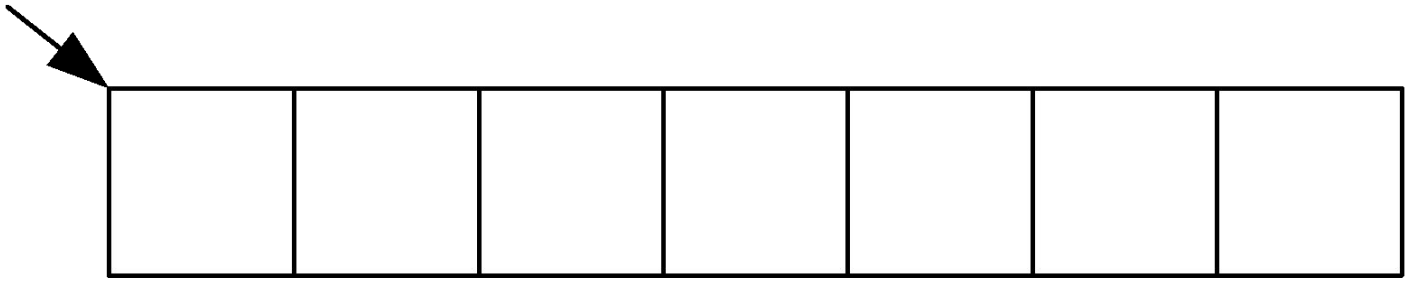
```
const int n = 100;
const int m = 20;
//с помощью именованных констант
short mas1[n], mas2[m]; // описание нескольких массивов
//с помощью константных выражений
long mas3[(n - m + 1)*2];
```

Полная инициализация значений ячеек массива

тип имя_массива [размер_массива] = {список_значений_инициализаторов}

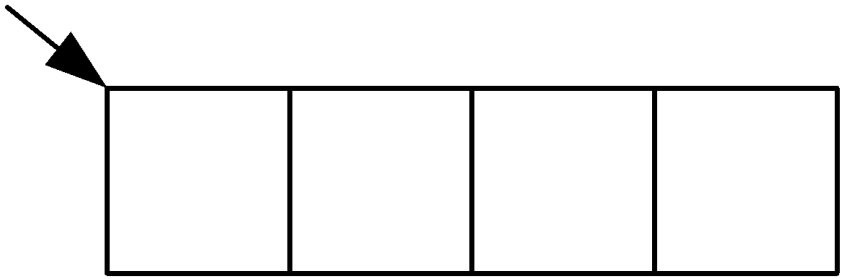
Например:

```
int mas [7] = {3, 5, 2, 6, 4, 2, 3};
// float b[2] = {1, 2, 3}; - ошибка, значения не помещаются
```



При полной инициализации размер массива можно не указывать

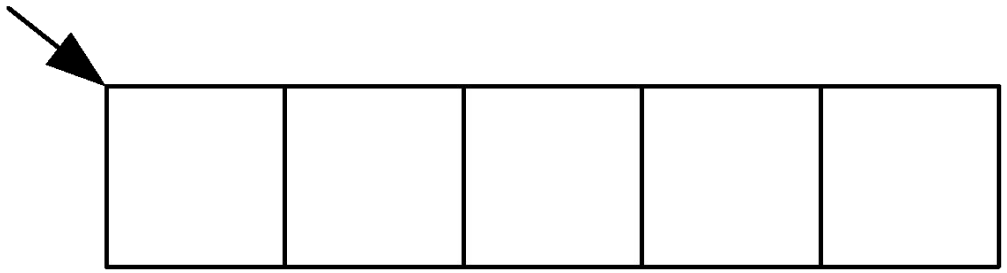
```
/*число ячеек массива определяется автоматически по количеству инициализаторов*/
bool flags[] = {true, false, true, true};
```



```
int x = 30;
int iArr[] = {10, 20, x};
/*выражения в списке инициализации не обязаны быть известными на стадии компиляции*/
```

Частичная инициализация значений ячеек массива

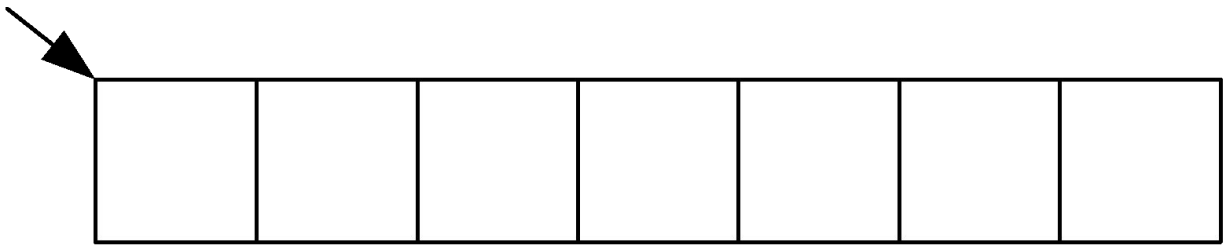
```
//в не проинициализированные ячейки записывается ноль

float money[5] = {5, 2.7, 9.9};
```



Операция индексации (доступ к элементам массива)

Первая по номеру ячейка массива имеет индекс 0

```
int mas [7] = {3, 5, 2, 6, 4, 2, 3};
//запись в переменную b значения из 4-й ячейки массива
int b = mas[3];
cout << b;
//замена значения в 1-й ячейке массива нулем
mas[0] = 0;
```



Ввод и вывод элементов массива

```
const int n = 7, m = 5;
int mas[n] = {3, 5, 2, 6, 4, 2, 3};
float a[m];
for (int i = 0; i < m; i++)
   cin >> a[i];
for (int i = 0; i < n; i++)
/*выводим элементы массива mas в столбик*/
   cout << mas[i] << "\n";
for (int i = 0; i < m; i++)
   /*выводим элементы массива a в строчку через пробел*/
   cout << a[i] << " ";
```

Заполнение массива последовательностью целых чисел

```
//заполнить массив значениями от 1 до n
for(int i = 0; i < n; i++)
   a[i] = i + 1;

//заполнить массив значениями кратными 5 от 5 до 100
for(int i = 0; i < n; i++)
   a[i] = (i + 1) * 5;
```

Заполнение массива случайными числами

```
#include <cstdlib> // для rand
#include <iostream>
using namespace std;

int main(){
const int n = 7, m = 5;
int mas[n];
float a[m];
for (int i = 0; i < n; i++)
   mas[i] = rand() % 1000;          //целые числа [0;999]
   //mas[i] = rand() % 1000 - 500; //[-500; 499]

for (int i = 0; i < m; i++)
   a[i] = rand() % 1000 / 10.0; //дробные [0.0;100.0)
                    //с 1 знаком после запятой
}
```

Задача: определить по номеру месяца количество дней в нем (считать, что год не високосный)

```
int month;
cin >> month;
int days[12] = {31, 28, 31, 30, 31, 30,
          31, 31, 30, 31, 30, 31};
cout << days[month - 1];
```

Задача: вычислить сумму всех элементов массива

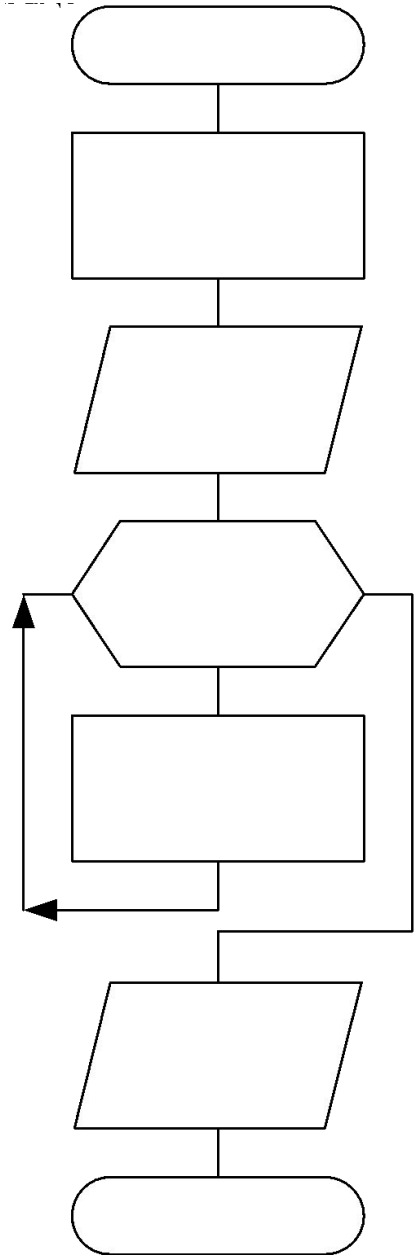
```
const int n = 10;
int mas[n], sum = 0;
for (int i = 0; i < n; i++)
   cin >> mas[i];
for (int i = 0; i < n; i++)
   sum += mas[i];
cout << sum;
```

Входные данные:

mas – массив из 10 целых чисел.

Выходные данные:

sum – целое число.



Задача: вычислить сумму элементов массива, кратных трем

```
const int n = 10;
int mas[n],
    sum = 0;
for (int i = 0; i < n; i++){
    cin >> mas[i];
    if (! (mas[i] % 3) )
        sum += mas[i];
    }
cout << sum;
```

Задача: заменить значение 2-й ячейки произведением всех элементов массива

```
const int n = 10;
float mas[n],
    prod = 1;
for (int i = 0; i < n; i++)
   cin >> mas[i];
for (int i = 0; i < n; i++)
   prod *= mas[i];
mas[1] = prod;
for (int i = 0; i < n; i++)
   cout << mas[i] << ' ';
```

Задача: найти позицию минимального элемента в массиве

```
const int n = 10; int k = 0;
float mas[n], minVal;
for (int i = 0; i < n; i++)
   cin >> mas[i];
minVal = mas[0];
for (int i = 1; i < n; i++)
   if (mas[i] < minVal){ //if (mas[i] < mas[k])
      minVal = mas[i];   //k = i;
      k = i;
   }
cout << k << ", value: " << mas[k];
```

*альтернативный способ поиска – хранить индекс минимального значения, а не само значение

Задача: наибольшая разность

За один обход найдите максимальную разность между двумя значениями массива. Массив состоит из неповторяющихся целых чисел от 1 до 100 включительно. Разрешено использовать только один цикл для обхода массива.

```
const int n = 10;
int arr[n] = {/*…*/};
int maxInd = 0, minInd = 0;
for (int i = 1; i < n; i++){
  if (arr[i] > arr[maxInd])
    maxInd = i;
  if (arr[i] < arr[minInd])
    minInd = i;
   }
cout << arr[maxInd] - arr[minInd];
```

Текущий контроль

Что будет выведено на экран после выполнения кода ниже

```
int mas[] = {10, 20, 30, 40, 50};
for (int i = 0; i <= 5; i++);
   cout << mas[i];
```

а) 1020304050
б) 10 20 30 40 50
в) 10203040500
г) 10 20 30 40 50 0
д) 1020304050 и случайное число
е) 10 20 30 40 50 и случайное число
ж) код не выполнится из-за ошибки компиляции

Текущий контроль

Что будет выведено на экран после выполнения кода ниже

```
int mas[] = {10, 20, 30, 40, 50};
for (int i = 0; i <= 5; i++);
   cout << mas[i];
```

а) 1020304050
б) 10 20 30 40 50
в) 10203040500
г) 10 20 30 40 50 0
д) 1020304050 и случайное число
е) 10 20 30 40 50 и случайное число
ж) код не выполнится из-за ошибки компиляции

Текущий контроль

Что будет выведено на экран после выполнения кода ниже

```
int mas[] = {10, 20, 30, 40, 50};
for (int i = 0; i <= 5; i++)
   cout << mas[i];
```

а) 1020304050
б) 10 20 30 40 50
в) 10203040500
г) 10 20 30 40 50 0
д) 1020304050 и случайное число
е) 10 20 30 40 50 и случайное число
ж) код не выполнится из-за ошибки компиляции

Текущий контроль

Что будет выведено на экран после выполнения кода ниже

```
int mas[] = {10, 20, 30, 40, 50};
for (int i = 0; i <= 5; i++)
   cout << mas[i];
```

а) 1020304050
б) 10 20 30 40 50
в) 10203040500
г) 10 20 30 40 50 0
д) 1020304050 и случайное число
е) 10 20 30 40 50 и случайное число
ж) код не выполнится из-за ошибки компиляции

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
#include <iostream>
int main(){
const int n = 7;
int mas [n] = {3, 5, 2, 6, 4, 1, 3};
for(int i = 0; i < 7; i += 2)
   mas[i]++;
std::cout << mas[4] - mas[5];
}
```

а) -1
б) 2
в) 3
г) 4
д) 1

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
#include <iostream>
int main(){
const int n = 7;
int mas [n] = {3, 5, 2, 6, 4, 1, 3};
for(int i = 0; i < 7; i += 2)
   mas[i]++;
std::cout << mas[4] - mas[5];
```

а) -1
б) 2
в) 3
г) 4
д) 1

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
const int n = 7;
int arr[n];
arr[1] = 3;
arr[7] = 5;
int s = 0;
for(int i = 0; i < n; i++)
   s += arr[i];
cout << s;
```

а) 0
б) 3
в) 5
г) 8
д) ошибка компиляции
е) непредсказуемый результат из-за выхода за границы массива

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
const int n = 7;
int arr[n];
arr[1] = 3;
arr[7] = 5;
int s = 0;
for(int i = 0; i < n; i++)
   s += arr[i];
cout << s;
```

а) 0
б) 3
в) 5
г) 8
д) ошибка компиляции
е) непредсказуемый результат из-за выхода за границы массива

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
const int n = 5;
int arr[n];
for(int i = 0; i < n; i++)
   cout << arr[i];
```

а) 00000

б) 0 0 0 0 0

в) пять «мусорных» значений

г) ничего, будет ошибка компиляции

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
const int n = 5;
int arr[n];
for(int i = 0; i < n; i++)
   cout << arr[i];
```

а) 00000

б) 0 0 0 0 0

в) пять «мусорных» значений

г) ничего, будет ошибка компиляции

Синтаксические возможности языка C++: массив констант

```
const int m[3] = {1, 2, 3}; //инициализация обязательна
//попытка изменения значения отслеживается компилятором
```

Синтаксические возможности языка C++: создание псевдонима статического массива

```
typedef int IntArray10[10];
```

или (только начиная с C++11)

```
using IntArray10 = int[10];
```

далее можно использовать

```
IntArray10 a, b;
```

то же самое что

```
int a[10], b[10];
```

Синтаксические возможности языка C++: оператор sizeof и статические массивы

```
int a;
int b[10];
cout << sizeof(a) //4
   << " "
      << sizeof(b);//40
```

Синтаксические возможности языка: нововведения C++11

В C++11 в стандартной библиотеке появились шаблоны функций begin() и end().

Это позволяет использовать массивы в диапазонном цикле for.

```
int arr[5];
for (auto element : arr)
   cout << element << " ";
```

Синтаксические возможности языка: нововведения C++14

В стандарте C++14 о константности размера массива не упоминается и можно использовать запись типа:

```
int n;
cin >> n;
int a[n];
```

где n – переменная.

Следует помнить, что это **не стандарт**, просто **компилятор gcc** дает расширение для поддержки массивов переменной длины.

Синтаксические возможности языка: нововведения C++17

В C++17 появился шаблон функции size()

```
int arr[4];
cout << size(arr) << '\n';
```

Контейнер array в стандартной библиотеке шаблонов (stl)

```
#include <array>
using namespace std;

int main(){
array <int, 5> arrInt5; //GARBAGE
array <int, 10> arrInt10 = {1, 2, 3}; //default values
arrInt5.fill(0);
cout << arrInt10.size() << endl;;
int x = arrInt5[10]; // undefined behavior
arrInt10.back() += 5;
cout << arrInt10.front() << endl;
for (auto element : arrInt10)
   cout << element << " ";
}
```

Универсальная инициализация

В C++11 появилась универсальная инициализация

```
int arr[10]{1, 2, 3};
array <int, 10> arr2 {4, 5, 6, 7};
```

Текущий контроль

Какой из способов определения массива является допустимым в языке C++?

а) int mas[5];

б) int mas(5);

в) int mas[] = {1, 2, 3};

г) int mas() = {1, 2, 3};

д) int mas[5] = {1, 2, 3};

е) int mas[5] {1, 2, 3};

ж) int mas[] {1, 2, 3};

з) int mas(5) {1, 2, 3};

и) int mas[2] {1, 2, 3};

к) int mas[2] {1, 2, };

Текущий контроль

Какой из способов определения массива является допустимым в языке C++?

а) int mas[5];

б) int mas(5);

в) int mas[] = {1, 2, 3};

г) int mas() = {1,2,3};

д) int mas[5] = {1,2,3};

е) int mas[5] {1,2,3};

ж) int mas[] {1,2,3};

з) int mas(5) {1,2,3};

и) int mas[2] {1,2,3};

к) int mas[2] {1, 2, };

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
const int n = 5;
int arr[n]{};
for(int i = 0; i < n; i++)
   cout << arr[i];
```

а) 00000

б) 0 0 0 0 0

в) пять «мусорных» значений

г) ничего, будет ошибка компиляции

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
const int n = 5;
int arr[n]{};
for(int i = 0; i < n; i++)
   cout << arr[i];
```

а) 00000

б) 0 0 0 0 0

в) пять «мусорных» значений

г) ничего, будет ошибка компиляции

Текущий контроль

Для массива, определенного следующим образом, как правильно должно выглядеть обращение к последнему элементу
int mas[12];

а) mas[12];
б) mas[11];
в) mas[length(mas)];
г) mas[-1];
д) mas[length(mas) - 1];
е) mas[size(mas)];
ж) mas[size(mas) - 1];
з) mas.front();
и) mas.back();

Текущий контроль

Для массива, определенного следующим образом, как правильно должно выглядеть обращение к последнему элементу
int mas[12];

а) mas[12];
б) mas[11];
в) mas[length(mas)];
г) mas[-1];
д) mas[length(mas) - 1];
е) mas[size(mas)];
ж) mas[size(mas) - 1];
з) mas.front();
и) mas.back();

Текущий контроль

Для массива, определенного следующим образом, как правильно должно выглядеть обращение к последнему элементу
array <int, 12> mas;

а) mas[12];
б) mas[11];
в) mas[-1];
г) mas[mas.size()];
д) mas[mas.size() - 1];
е) mas.front();
ж) mas.back();

Текущий контроль

Для массива, определенного следующим образом, как правильно должно выглядеть обращение к последнему элементу
array <int, 12> mas;

а) mas[12];
б) mas[11];
в) mas[-1];
г) mas[mas.size()];
д) mas[mas.size() - 1];
е) mas.front();
ж) mas.back();

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
int s = 0;
int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
for(int i = 0; i <= sizeof(arr) / sizeof(int) / 2; i++)
   if(i % 3 == 2)
      s += arr[i];
cout << s;
```

а) 2
б) 3
в) 7
г) 9
д) 15
е) 18

Текущий контроль

Что будет выведено на экран в результате работы программы?

```
int s = 0;
int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
for(int i = 0; i <= sizeof(arr) / sizeof(int) / 2; i++)
   if(i % 3 == 2)
      s += arr[i];
cout << s;
```

а) 2
б) 3
в) 7
г) 9
д) 15
е) 18

Пример решения задач лабораторной работы

Задача 1. Дан массив из 10 различных целых чисел. Заменить первый элемент массива на сумму только тех его чисел, которые принадлежат промежутку от *a* до *b*;

Например, для массива чисел: 1 3 7 2 0 6 5 4 9 8, a = 3, b = 7 будем рассматривать только следующие значения:

1 3 7 2 0 6 5 4 9 8 □ s = 25

Итоговый массив: 25 3 7 2 0 6 5 4 9 8

Задача 2. Дан массив из 10 действительных чисел и натуральные числа *k* и t. Вычислить сумму элементов массива, расположенных между *k*-м и t-м его элементами.

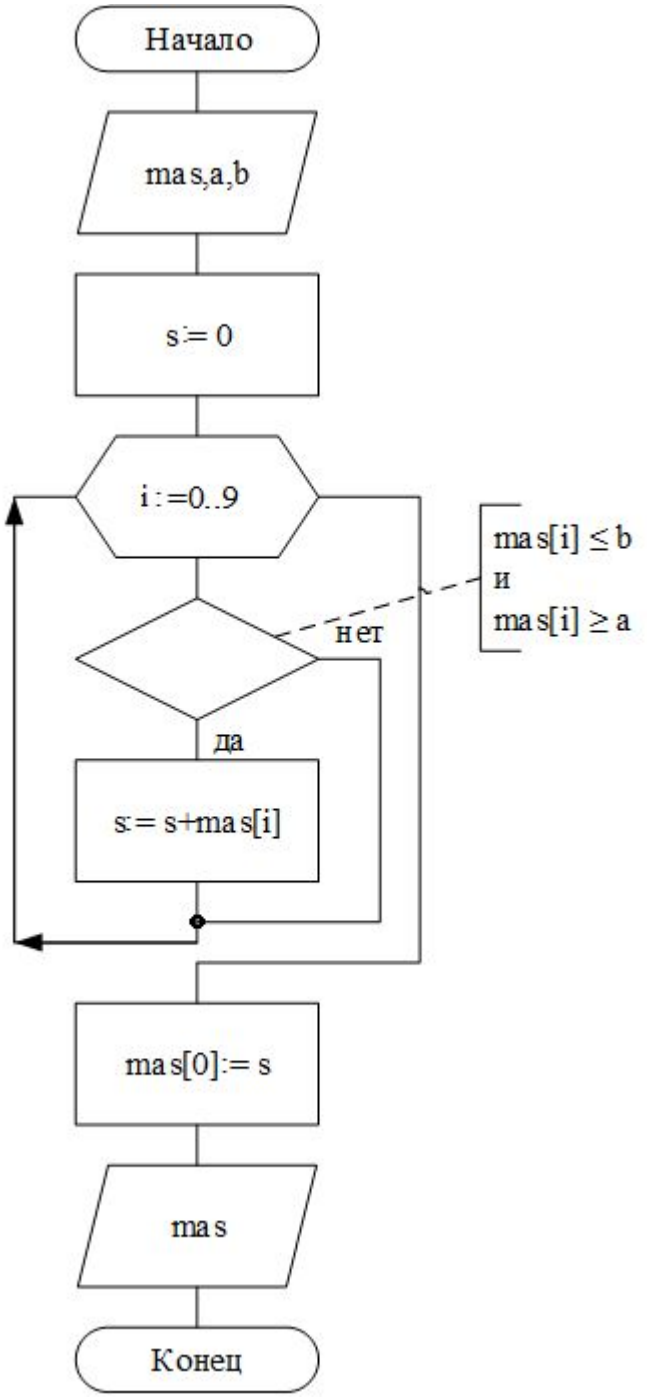
При решении задачи необходимо проверить расположение t относительно k и делать вычисления либо от k+1 до t-1, либо от t+1 до k-1

Предполагается, что пользователь вводит порядковые номера элементов (не индексы)

Решение задачи 1

Входные данные:
mas – массив из 10 целых чисел
a, b - целые числа
Выходные данные:
mas – массив из 10 целых чисел

```
short mas[10];
for (int i = 0; i < 10; i++)
  cin >> mas[i];
short a, b, s = 0;
cin >> a >> b;
for (int i = 0; i < 10; i++)
  if ((mas[i] <= b) && (mas[i] >= a))
    s += mas[i];
mas[0] = s;
for (int i = 0; i < 10; i++)
  cout << mas[i];
```



Решение задачи 2

Входные данные:
mas – массив из 10 действительных чисел
Выходные данные:
s – действительное число

```
float mas[10];
for (int i = 0; i < 10; i++)
   cin >> mas[i];
int k, t;
cin >> k >> t;
float s = 0;
if (k > t)
   swap(k, t);
for (int i = k; i <= t - 2; i++)
   s += mas[i];
cout << s;
```