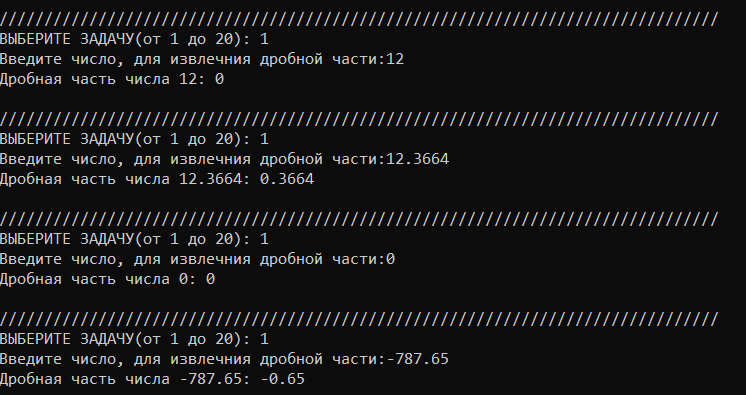
## **Задача 1: Дробная часть**

**Формулировка:** Дана сигнатура функции: double fraction (double x); Необходимо реализовать функцию таким образом, чтобы она возвращала только дробную часть числа х. Подсказка: вещественное число может быть преобразовано к целому путем отбрасывания дробной части.

**Что делает:**  Отделяет дробную часть от целой

**Алгоритм:**

* Берем число, отбрасываем всё после запятой - получаем целую часть
* Из исходного числа вычитаем целую часть - получаем дробную
* Возвращаем дробную часть

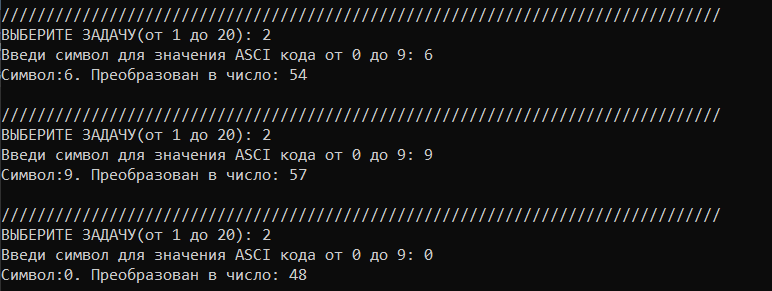
**Тесты:**

## **Задача 2:Букву в число**

## **Формулировка:** Дана сигнатура функции: int charToNum (char x); Функция принимает символ х, который представляет собой один из “0 1 2 3 4 5 6 7 8 9”. Необходимо реализовать функцию таким образом, чтобы она преобразовывал **Что делает:** Превращает символ цифры в число

**Алгоритм:**

* Получаем на вход символ
* Проверяем, что это действительно цифра от 0 до 9
* Возвращаем его ASCII код (сейчас баг - должен возвращать число, а не код

**Тесты:**

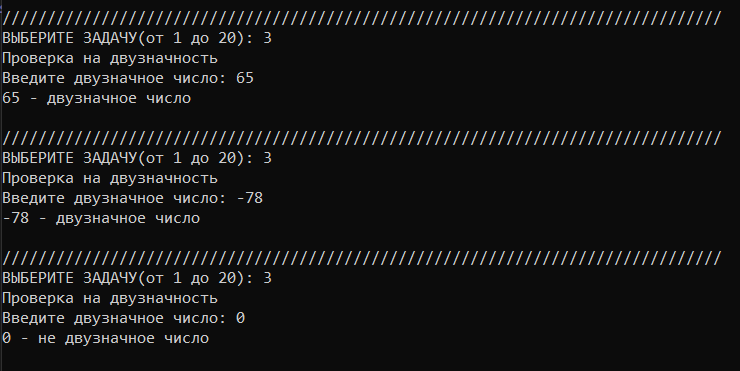
## **Задача 3: Двузначное**

**Формулировка:** Дана сигнатура функции: bool is2Digits (int x); Необходимо реализовать функцию таким образом, чтобы она принимала число x и возвращала true, если оно двузначное.

**Что делает:** Проверяет, двузначное ли число

**Алгоритм:**

* Смотрим на число
* Если оно от 10 до 99 ИЛИ от -99 до -10 - значит двузначное
* Иначе - не двузначное

**Тесты:**

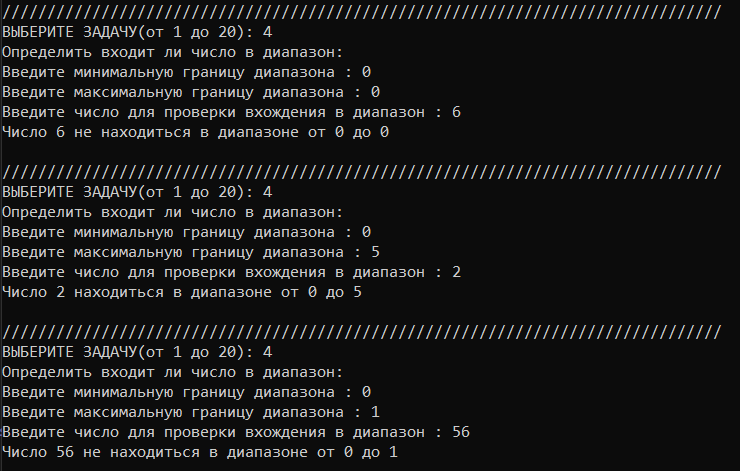
## **Задача 4: Диапазон**

**Формулировка:** Дана сигнатура функции: bool isInRange (int a, int b, int num); Функция принимает левую и правую границу (a и b) некоторого числового диапазона. Необходимо реализовать функцию таким образом, чтобы она возвращала true, если num входит в указанный диапазон (включая границы). Обратите внимание, что отношение a и b заранее неизвестно (неясно кто из них больше, а кто меньше)

**Что делает:** Проверяет, попадает ли число между двумя другими

**Алгоритм:**

* Получаем два числа - границы диапазона
* Определяем, какое из них меньше, какое больше
* Смотрим, находится ли наше число между ними
* Если да - true, если нет - false

**Тесты:**

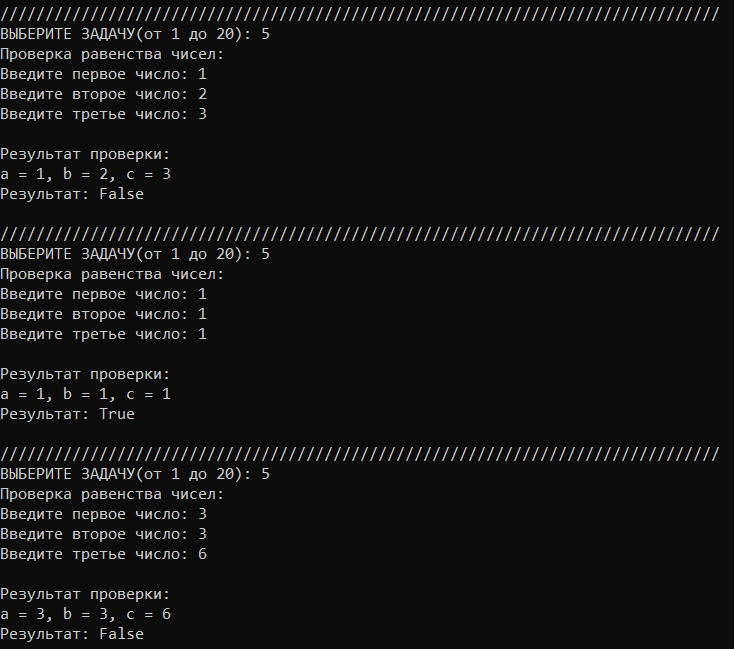
## **Задача 5: Равенство**

**Формулировка:** Дана сигнатура функции: bool isEqual(int a, int b, int c); Необходимо реализовать функцию таким образом, чтобы она возвращала true, если все три полученных функцией числа равны

**Что делает:** Проверяет, все ли три числа одинаковые

**Алгоритм:**

* Берем три числа
* Проверяем, равны ли первое второму И второе третьему
* Если оба условия true - все числа равны

**Тесты:**

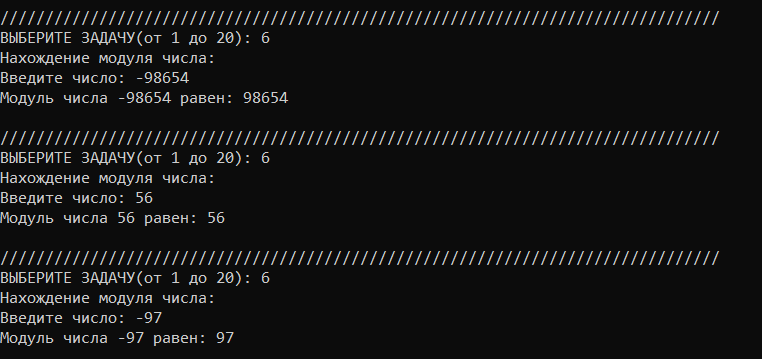
## **Задача 6: Модуль числа**

**Формулировка:** Дана сигнатура функции: int abs (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала модуль числа х (если оно было положительным, то таким и остается, если он было отрицательным – то необходимо вернуть его без знака минус).

**Что делает:** Находит абсолютное значение числа

**Алгоритм:**

* Если число положительное - возвращаем как есть
* Если отрицательное - убираем минус

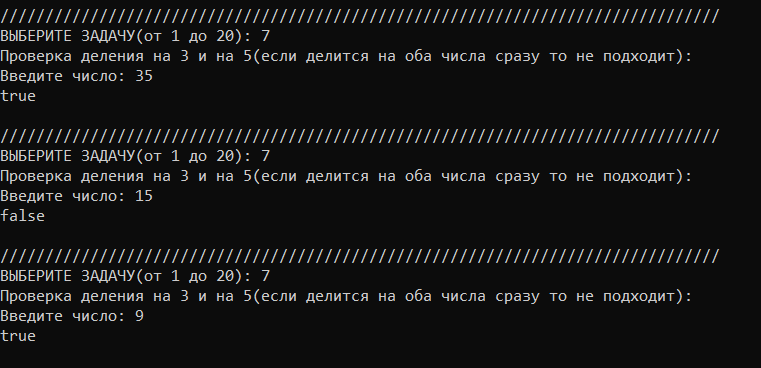
**Тесты:**

## **Задача 7: Тридцать пять**

**Формулировка:** Дана сигнатура функции: bool is35 (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала true, если число x делится нацело на 3 или 5. При этом, если оно делится и на 3, и на 5, то вернуть надо false. Подсказка: оператор % позволяет получить остаток от деления.

**Что делает:** Проверяет делится ли число на 3 ИЛИ на 5, но не на оба сразу **Алгоритм:**

* Проверяем делимость на 3
* Проверяем делимость на 5
* Если делится на один ИЛИ на другой, но НЕ на оба - true
* Иначе - false

**Тесты:**

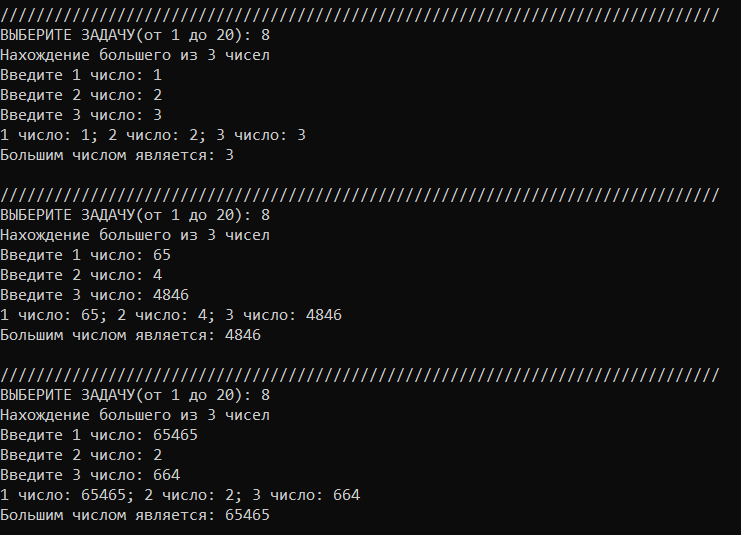
## **Задача 8: Тройной максимум**

**Формулировка:** Дана сигнатура функции: int max3 (int x, int y, int z); Необходимо реализовать функцию таким образом, чтобы она возвращала максимальное из трех полученных функцией чисел. Подсказка: идеальное решение включает всего две инструкции if и не содержит вложенных if.

**Что делает:** Находит самое большое из трех чисел

**Алгоритм:**

* Предполагаем, что первое число - максимальное
* Сравниваем со вторым, если второе больше - запоминаем его как максимальное
* Сравниваем с третьим, если третье больше - запоминаем его
* Возвращаем самое большое

**Тесты:**

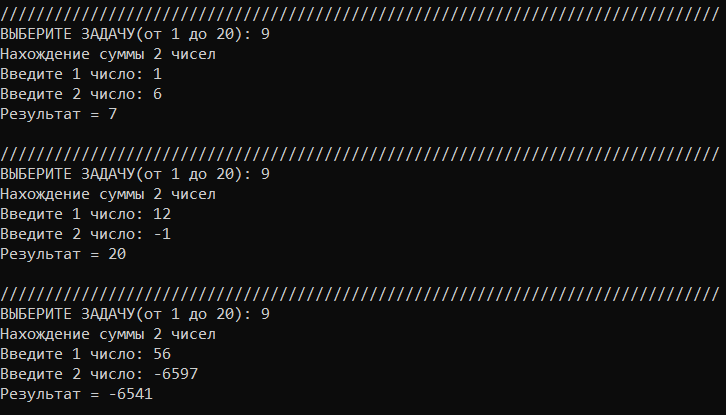
## **Задача 9: Двойная сумма**

**Формулировка:** Дана сигнатура функции: int sum2 (int x, int y); Необходимо реализовать функцию таким образом, чтобы она возвращала сумму чисел x и y. Однако, если сумма попадает в диапазон от 10 до 19, то надо вернуть число 20.

**Что делает:**  Складывает два числа с особым условием

**Алгоритм:**

* Складываем два числа
* Если сумма от 10 до 19 - возвращаем 20
* Иначе возвращаем обычную сумму

**Тесты:**

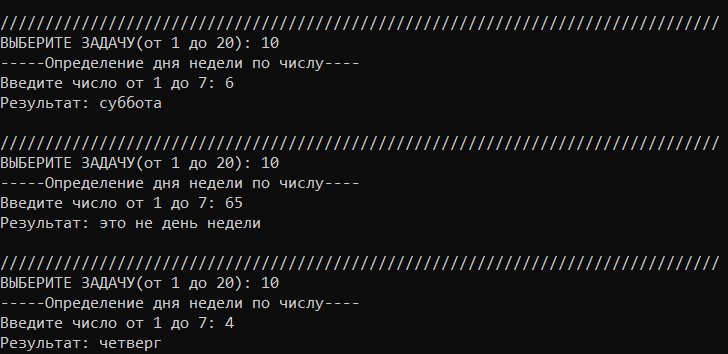
**Задача 10: День недели**

**Формулировка:** Дана сигнатура функции: String day (int x); Функция принимает число x, обозначающее день недели. Необходимо реализовать функцию таким образом, чтобы она возвращала строку, которая будет обозначать текущий день недели, где 1 — это понедельник, а 7 – воскресенье. Если число не от 1 до 7 то верните текст “это не день недели”. Вместо if в данной задаче используйте switch.

**Что делает:** По номеру возвращает название дня недели

**Алгоритм:**

* Получаем число от 1 до 7
* Для каждого числа возвращаем соответствующее название дня
* Если число не от 1 до 7 - сообщаем об ошибке

**Тесты:**

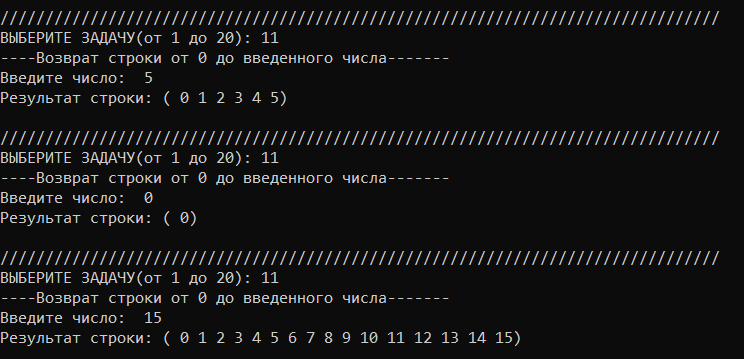
**Задача 11: Числа подряд**

**Формулировка:** Дана сигнатура функции: String listNums (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала строку, в которой будут записаны все числа от 0 до x (включительно).

**Что делает:**  Создает строку с числами от 0 до заданного

**Алгоритм:**

* Начинаем с 0
* Добавляем каждое следующее число до нужного
* Между числами ставим пробелы
* Возвращаем готовую строку

**Тесты:**

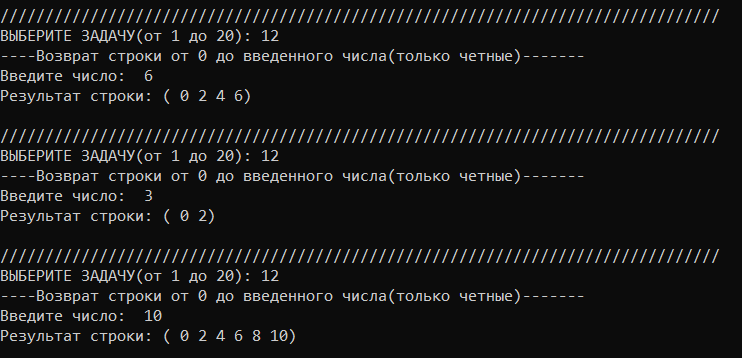
**Задача 12: Четные числа**

**Формулировка:** Дана сигнатура функции: String chet (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала строку, в которой будут записаны все четные числа от 0 до x (включительно). Подсказа для обеспечения качества кода: инструкцию if использовать не следует.

**Что делает:** Создает строку с четными числами от 0 до заданного

**Алгоритм:**

* Начинаем с 0
* Перебираем числа с шагом 2 (только четные)
* Добавляем их в строку через пробелы
* Возвращаем строку

**Тесты:**

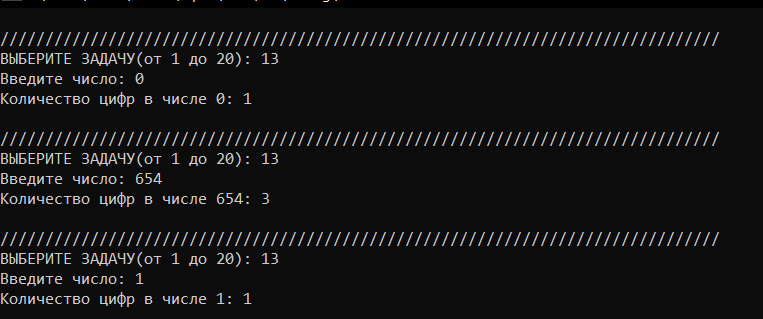
**Задача 13: Длина числа**

**Формулировка:** Дана сигнатура функции: int numLen (long x); Необходимо реализовать функцию таким образом, чтобы она возвращала количество знаков в числе x. Подсказка: Int у=123/10; // у будет иметь значение 12

**Что делает:**  Считает сколько цифр в числе

**Алгоритм:**

* Если число 0 - сразу возвращаем 1
* Если отрицательное - убираем минус
* Делим число на 10 пока не получим 0
* Считаем сколько раз поделили - это и есть количество цифр

**Тесты:**

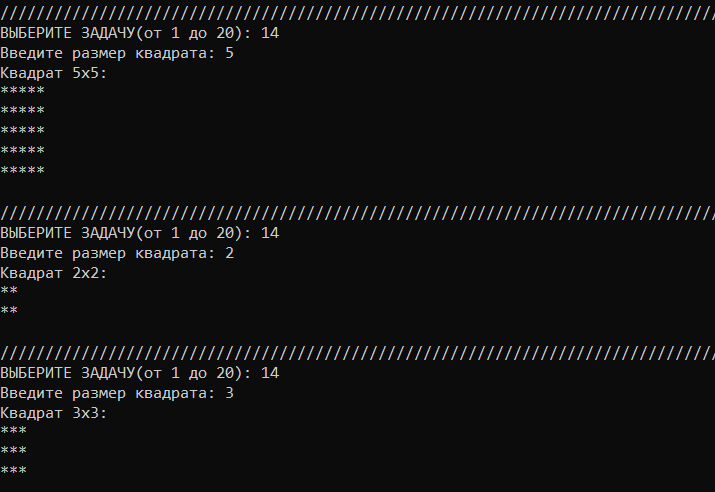
## **Задача 14: Квадрат**

**Формулировка:** Дана сигнатура функции: void square (int x); Необходимо реализовать функцию таким образом, чтобы она выводила на экран квадрат из символов ‘\*’ размером х, у которого х символов в ряд и х символов в высоту.

**Что делает:** Рисует квадрат в консоли

**Алгоритм:**

* Для каждой строки от 0 до размера квадрата:
* Для каждой колонки от 0 до размера квадрата:
* Печатаем звездочку
* В конце строки переходим на новую

**Тесты:**

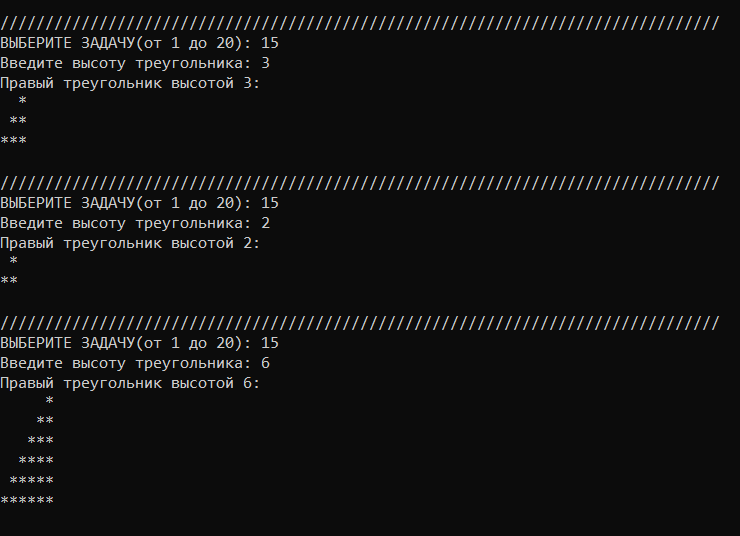
## **Задача 15: Правый треугольник**

**Формулировка:** Дана сигнатура функции: void rightTriangle (int x); Необходимо реализовать функцию таким образом, чтобы она выводила на экран треугольник из символов ‘\*’ у которого х символов в высоту, а количество символов в ряду совпадает с номером строки, при этом треугольник выровнен по правому краю. Подсказка: перед символами ‘\*’ следует выводить необходимое количество пробелов.

**Что делает:** Рисует треугольник выровненный по правому краю

**Алгоритм:**

* Для каждой строки от 1 до высоты:
* Сначала печатаем пробелы: (общая высота - номер строки)
* Потом печатаем звездочки: количество = номер строки
* Переходим на новую строку

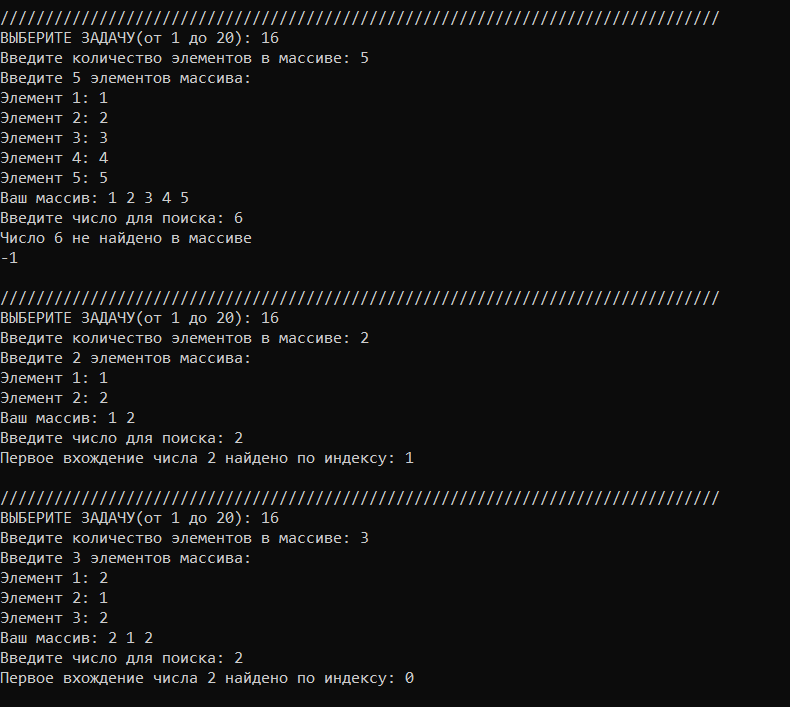
**Тесты:**

## **Задача 16: Поиск первого значения**

**Формулировка:** Дана сигнатура функции: int findFirst (int arr[], int x); Необходимо реализовать функцию таким образом, чтобы она возвращала индекс первого вхождения числа x в массив arr. Если число не входит в массив – возвращается -1.

**Что делает:** Ищет первое вхождение числа в массиве **Алгоритм:**

* Проходим по всем элементам массива от начала до конца
* Сравниваем каждый элемент с искомым числом
* Если нашли совпадение - возвращаем индекс
* Если дошли до конца и не нашли - возвращаем -1

**Тесты:**

## **Задача 17: Поиск максимального**

**Формулировка:** Дана сигнатура функции: int maxAbs (int arr[]); Необходимо реализовать функцию таким образом, чтобы она возвращала наибольшее по модулю (то есть без учета знака) значение массива arr.

**Что делает:** Находит число с самым большим абсолютным значением

**Алгоритм:**

* Если массив пустой - возвращаем 0
* Предполагаем, что первый элемент - максимальный
* Проходим по всем остальным элементам:
* Сравниваем модуль текущего элемента с модулем максимального
* Если текущий больше - запоминаем его как новый максимальный
* Возвращаем найденное число

**Тесты:**