

Тестирующая система расположена по адресу:

<http://ejudge.cs.msu.ru/>

По тому же адресу расположена электронная версия данной памятки.

Во всех задачах можно использовать как файловый ввод-вывод, так и ввод с клавиатуры и вывод на экран.

## Pascal

Для считывания данных можно использовать функции `read` и `readln`.

```
var n, b, c, i : Integer;
    a : array [1 .. 100] of Longint;
    s : String;
begin
    // Считать одно число никогда не было проблемой:
    read(n);
    // Считать два числа тоже совсем несложно:
    read(b, c); // числа при этом могут быть разделены пробелом или переводом строки
    for i := 1 to n do begin
        read(a[i]); // массив можно считывать последовательно по одному числу
    end;
    // Считать строку до конца строки
    readln(s);
    // Если мы попытаемся считать следующую строку после команды read
    read(a, b); // 1 2
    readln(s); // abracadabra, в отдельной строке
    // мы столкнемся с проблемой: строка будет пустой.
    // Причина состоит в том, что строка не была переведена,
    // то есть мы продолжаем считывать из той же строки, хотя данные в ней закончились.
    // Решением является принудительный перевод строки, используя команду
    readln;
    // Существует также сокращенный вариант, который считывает данные и перевод строки:
    readln(a, b);
    readln(s);
end.
```

## C/C++

Для считывания данных можно использовать функцию `scanf`, предварительно подключив библиотеку `#include <stdio.h>`:

```
#include <stdio.h>
int main() {
    int n, b, c;
    // Для считывания чисел нужно использовать спецификатор %d
    scanf("%d", &n); // не забудьте про знак &
    // С помощью этой функции можно считывать несколько чисел, числа при этом
    // могут быть разделены пробелом или переводом строки
    scanf("%d %d", &b, &c);
    int a[100], i;
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]); // можно считывать элементы последовательности
    // Для считывания строк нужно использовать спецификаатор %s,
    // нижеследующая команда считывает очередную часть строки до первого пробельного символа
    char str[100];
    scanf("%s", str); // а здесь знак & не нужен!
    // Для считывания до конца строки следует использовать
    // специальную функцию fgets с тремя аргументами.
    // Первый аргумент - строка, второй - максимальная ожидаемая длина,
    // третий - файловый дескриптор (stdin соответствует вводу с клавиатуры)
    fgets(str, 100, stdin);
}
```

```

    return 0;
}

```

В языке C++ можно также использовать потоки ввода `cin/cout`, предварительно подключив библиотеку `#include <iostream>`. Обращаем ваше внимание, `cin/cout` работают значительно медленнее, чем `scanf/printf`.

```

#include <iostream>
#include <string>
int main() {
    int n, b, c, a[100];
    std::cin >> n >> b >> c;
    for (int i = 0; i < n; ++i)
        std::cin >> a[i];
    std::string s;
    std::cin >> s; // до пробельного символа
    getline(std::cin, s); // до конца строки
    return 0;
}

```

## Python

Для обработки данных в программах на языке Python3 используется чтение строк и дальнейшее их преобразование:

```

# Считывание строки до конца строки
a = input()
# Считывание целого числа
# (работает, если в строке содержится только одно число и больше ничего!)
b = int(input())
# Если же мы хотим считать два числа, разделенные каким-то
# количеством пробельных символов (но не переводом строки!),
# можно воспользоваться конструкцией:
c, d = map(int, input().split())
# Аналогично для трех и более чисел:
c, d, e = map(int, input().split())
# Если же мы хотим считать последовательность чисел,
# которые расположены в одной строке, нужно использовать массивы:
f = list(map(int, input().split()))
# Теперь f - это массив (list) чисел, расположенных в соответствующей строке

```

Обращаем ваше внимание, что зачастую под интерпретатором PyPy (руру и руру3.0 в тестирующей системе) программы выполняются быстрее, чем под интерпретатором CPython (python и python3 в тестирующей системе).

## Java

Для ввода на языке Java можно воспользоваться очень удобным классом `java.util.Scanner`:

```

import java.util.Scanner;
public class Solution {
    public static void main(String[] args) {
        new Solution().run();
    }
    public void run() {
        Scanner in = new Scanner(System.in);
        // Считать число совсем просто:
        int n = in.nextInt();
        // Не имеет значения, в одной строке записаны числа (через пробел) или в разных строках
        int a = in.nextInt(), b = in.nextInt();
        String s = in.next(); // Считать строку до ближайшего пробельного символа
        String t = in.nextLine(); // Считываем до конца строки
    }
}

```