

### Раздел «Язык Си» . MaximumProblem :

#### Максимум

Напишите программу, которая находит максимум введенных целых чисел. На вход поступает количество чисел, а затем сами числа.

Вот решение этой задачи:

```
/* Программа max.c "Максимум чисел" */
#include <stdio.h>
int main ()
{
    int i, n, a, max;
    printf ("Введите количество чисел: ");
    scanf ("%d", &n);
    printf ("Введите %d чисел: ", n);
    scanf ("%d", &max);
    for(i = 1; i < n ; i++)
    {
        scanf ("%d", &a);
        if(a > max)
            max = a;
    }
    printf ("%d", max);
    return 0;
}
```

Числа вы можете вводить, разделяя их пробелом SPACE, символом табуляции TAB, или нажимая после каждого введенного числа ENTER. Символы SPACE, TAB, ENTER называются пробельными символами (white space). Функция scanf считывает объекты, разделенные любым числом пробельных символов.

Обратите внимание на второй printf – он имеет два аргумента:

```
printf ("Введите %d чисел: ", n);
```

Первый аргумент – это строка (текст в двойных кавычках), которая задает формат того, что будет печататься. В этой строке встречается выражение %d, которое соответствует формату "десятичная запись целого числа". На месте этого выражения будет напечатано на экране компьютера значение второго аргумента, то есть значение переменной n.

В программе max.c мы впервые встречаемся с *условным оператором if* и *оператором цикла for*.

Оператор условного перехода записывается так:

```
if(A) B;
```

Он соответствует предложению "Если выполнено условие A, то сделать B".

Оператор for устроен следующим образом:

```
for(A; B; C) D;
```

Элемент D может быть как одной командой, так и произвольным набором команд, заключенных в *блок*.

Поиск

Поиск

Раздел «Язык Си»

Главная  
Зачем учить C?  
Определения

Инструменты:

Поиск  
Изменения  
Index  
Статистика

Разделы

Информация  
Алгоритмы  
Язык Си  
Язык Ruby  
Язык  
Ассемблера  
E! Judge  
Парадигмы  
Образование  
Сети  
Objective C

Logon>>

Команды объединяются в блок с помощью заключения их в фигурные скобки. В нашем случае D это

```
scanf ("%d", &a);
if(a > max) max = a;
```

Элемент D называется телом цикла – это то, что будет выполняться несколько раз. Сколько именно? Это зависит от B – тело D будет выполняться пока выполнено условие B.

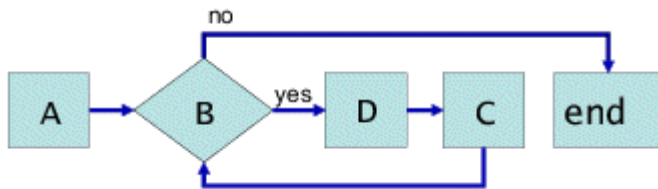
Например, строка

```
for( i = 0 ; i < 10 ; i++ ) { printf ( "Hi!" ); }
```

означает "10 раз напечатать слово Hi! ". Более подробно:

1. положим  $i = 0$ ;
2. проверим, что  $i < 10$ ; если условие не выполнено, то перейдем к пункту 5;
3. выполним `printf ( "Hi!" );`;
4. выполним `i++` (увеличим переменную  $i$  на 1) и перейдем к пункту 2;
5. конец цикла;

Логiku оператора for можно изобразить в виде блок схемы:



Логика нашей программы следующая:

1. Объявляем все переменные, которые встретятся в нашей программе:
  - $n$  количество чисел,
  - $i$  переменная цикла,
  - $max$  переменная для хранения текущего найденного максимума,
  - $a$  переменная с очередным числом, в неё мы считываем каждое число.
2. Печатаем на экран приглашение ввести число  $n$ .
3. Считываем  $n$ .
4. Печатаем на экран приглашение "Введите  $n$  чисел".
5. Считываем первое из этих  $n$  чисел в переменную  $max$ .
6. В цикле  $n - 1$  раз считываем очередное число, и если оно больше, чем текущий максимум, то обновляем переменную  $max$ .
7. Выводим значение переменной  $max$ .

Вы уже наверное заметили, что когда мы в `scanf` указываем переменные, в которые нужно поместить считываемые данные, то перед именем переменной ставим знак `&` (амперсанд), в то время как в `printf` при печати значений переменных амперсанд не ставится. Операция "амперсанд" – это операция взятия адреса. Если  $a$  есть значение переменной  $a$ , то `&a` – это адрес в памяти компьютера, где хранится переменная  $a$ : одно есть само число, другое – место, где это число хранится.

- Функции `scanf` нужно знать не текущее значение переменной, а её адрес, чтобы положить туда результат считывания.
- Функции `printf` нужно знать значения переменных, чтобы их печатать.



Функции `scanf` нужно передавать **адреса** переменных, куда помещать считываемые данные.

(с) Материалы раздела "Язык Си" публикуются под лицензией GNU Free Documentation License.