Вычисление факториала числа

Задача

Факториал числа представляет собой произведение всех натуральных чисел от 1 до этого числа включительно. Например, факториал числа 7 выглядит так: 1 * 2 * 3 * 4 * 5 * 6 * 7

Факториал числа обозначается как само число после которого следует восклицательный знак. Например, 7!. Таким образом: 7! = 1 * 2 * 3 * 4 * 5 * 6 * 7 = 5040

С увеличением числа его факториал быстро возрастает. Так если 3! = 6, то уже 10! = 3628800. Поэтому для натуральных чисел больше 12-ти в языке программирования Паскаль просто так факториал вычислить нельзя.

Допустим, требуется определить факториал числа, которое ввел пользователь.

Решение

Переменной factorial сначала присваивается значение 1. 0! = 1 и 1! = 1.

Если пользователь ввел число больше единицы, то выполняется цикл, в теле которого на каждой итерации значение переменной `factorial` умножается на следующее натуральное число (переменную `i`).

Программа на языке Паскаль:

```
var
    factorial: longint;
    n, i: byte;

begin
    write('n = '); readln(n);

    factorial := 1;

    for i:=2 to n do
        factorial := factorial * i;

    writeln('n! = ', factorial);

readln;
end.
```

Посчитать четные и нечетные цифры числа

Задача
Написать программу, подсчитывающую количество четных и нечетных цифр в числе.
Решение

Описание переменных:

even - количество четных цифр;
odd - количество нечетных цифр.

Алгоритм решения задачи:

1. Если число делится без остатка на 2, значит последняя цифра четная (увеличиваем переменную even). Иначе - нечетная (тогда odd + 1).
2. Избавляемся от младшего разряда в числе: операция div на 10.

Программа на языке Паскаль:

```
var
        a: integer;
        even,odd: byte;
begin
        readln(a);
        even := 0;
        odd := 0;
        while a > 0 do begin
                if (a mod 2) = 0 then
                        even := even + 1
                else
                        odd := odd + 1;
                a := a div 10;
        end;
        writeln('even: ', even);
        writeln('odd: ', odd);
end.
```