

Программирование на языке Паскаль

**Тема. Подпрограммы.
Функции. Задачи с
рекурсивной формулировкой**

Рекурсивная функция — это такая функция, которая вызывает себя из себя же самой, но с другими значениями параметров.

Должна иметь условие остановки!



Задания

«А»: Напишите функцию, которая вычисляет факториал числа n ($n! = 1 * 2 * \dots * n$).

$$4! = 1 * 2 * 3 * 4$$


$$3! = 1 * 2 * 3$$


$$2! = 1 * 2$$

$$N! = (N-1)! * N$$


$$1!$$

Программа

```
function f (n: integer) :  
integer;  
var count,i: integer;  
begin  
    count := 1;  
    for i:= 1 to n do  
count := count*i;  
    f:= count;  
end;
```

```
function f (n: integer) :  
integer;  
  
begin  
if n=1 then f:= 1  
else f:= n*f(n-1);  
  
end;
```

Условие
остановки

Рекурсивная функция

Задания

«А»: Напишите функцию, которая вычисляет факториал числа n ($n! = 1 * 2 * \dots * n$). Найдите значение выражения: $1! + 2! + 3! + \dots + n!$

Пример: $n = 4$, **33**

«В»: Напишите функцию, которая определяет НОД (наибольший общий делитель) двух натуральных чисел.

Пример: Введите числа: 15 21, **НОД = 3**

«С»: Напишите функцию, которая вычисляет n число Фибоначчи (1, 1, 2, 3, 5, 8...).

Пример: Введите число: 6, **$n = 8$**

Программа «А»

```
var a, q, s: integer;  
function f (n: integer): integer;  
begin  
if n=1 then f:= 1  
else f:= n*f(n-1);  
end;
```

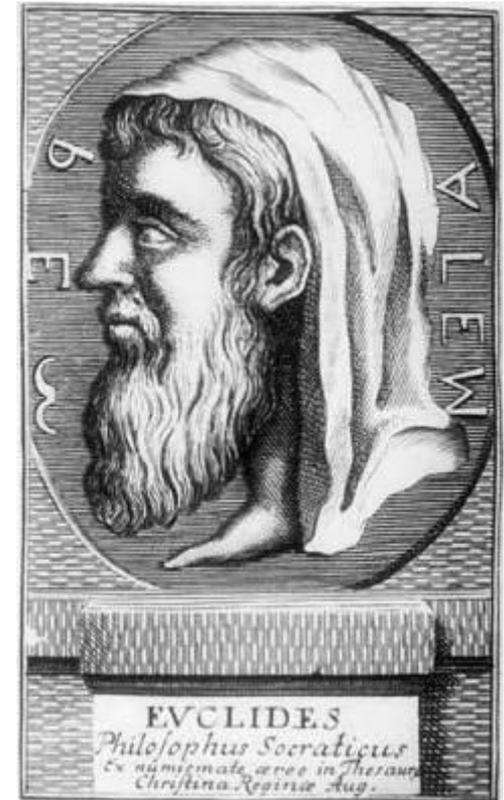
Рекурсивная функция

```
begin  
writeln('Введите число');  
read(a);  
For q:= 1 to a do s:=s+f(q);  
write(s);  
end.
```

Алгоритм нахождения наибольшего общего делителя (НОД) пары целых чисел (алгоритм Евклида)

Чтобы найти НОД двух чисел, составьте таблицу из двух столбцов и назовите столбцы X и Y . Запишите первое из заданных чисел в столбец X , а второе - в столбец Y . Если данные числа не равны, замените большее из них на результат вычитания из большего числа меньшего.

Повторяйте такие замены до тех пор, пока числа не окажутся равными, после чего число из столбца X считайте искомым результатом.



Программа «В»

```
var a, b, q: integer;
function NOD (a,b: integer): integer;
Begin
    if a=b then NOD:= a
        else if a>b then NOD:= NOD (a - b, b)
            else NOD:= NOD (a, b - a);
end;
begin
    writeln('Введите 2 числа');
    read(a, b);
    q:=NOD( a,b );
writeln('наибольший общий делитель = ', q);
end.
```

Условие остановки