

## Задание 11 – Рекурсивные алгоритмы

**Рекурсия** – это приём, позволяющий свести исходную задачу к одной или нескольким более простым задачам того же типа. Чтобы определить рекурсию, нужно задать:

- условие остановки рекурсии (базовый случай или несколько базовых случаев)
- рекуррентную формулу

Любую рекурсивную процедуру можно запрограммировать с помощью цикла. С другой стороны, рекурсия позволяет заменить цикл и в некоторых сложных задачах делает решение более понятным, хотя часто менее эффективным.

Примеры некоторых рекурсивных функций:

Функция находит произведение цифр числа в двоичном представлении:

```
function f(x: integer): integer;
begin
  if x > 1 then
    f := x mod 2 * f(x div 2)
  else
    f := x;
end;
```

Функция считает сумму цифр двоичной записи аргумента:

```
function f(x: integer): integer;
begin
  if x > 0 then
    f := x mod 2 + f(x div 2)
  else
    f := 0;
end;
```

Функция **f** возвращает сумму первой непрерывной последовательности нечетных цифр числа **x**, начиная с младшей:

```
function f(x: integer): integer;
begin
  if x mod 2 = 1 then
    f := x mod 10 + f(x div 10)
  else
    f := 0;
end;
```

Функция **F(x)** вычисляет факториал переданного ей числа, то есть произведение  $x! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (x-1) \cdot x$ :

```
function F(x: integer): longint;
begin
  if x = 1 then
    F := 1
  else F := x * F(x-1);
end;
```