

# Решение задач ЕГЭ по информатике по теме «Рекурсивный алгоритм».

Учитель МАОУ «Гимназия №1  
Октябрьского района г. Саратова»  
Зубашкова Татьяна Николаевна

# Задание 11

- **Умение исполнить  
рекурсивный алгоритм**
- **базовый уровень**
- **1 балл**

**Рекурсия** — это определение объектов через самих себя, вызов функции (процедуры) из неё же самой или через другие рекурсии.

- ✓ Рекурсия обычно используется тогда, когда в результате исходная задача сводится к более простой.
- ✓ Любую рекурсивную процедуру можно запрограммировать с помощью цикла.
- ✓ Рекурсия позволяет заменить цикл и в некоторых сложных задачах делает решение более понятным, хотя часто менее эффективным.

При решении задачи бывает необходимо разделять программу на отдельные части, которые называются подпрограммами.

*Подпрограмма - это повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы.*

Она записывается однократно, а в соответствующих местах программы обеспечивается лишь обращение к ней по имени.

**Подпрограммы** делятся на две категории: *процедуры и функции*.

Для удобства передачи данных в процедуру и получения из неё результата используются формальные и фактические параметры.

Формальные — условные обозначения в описании процедуры — описываются в её заголовке.

Фактические — с которыми требуется выполнить процедуру — перечисляются при вызове процедуры.

# Рекурсивные алгоритмы:

- Вызов рекурсивных процедур
- Алгоритмы, опирающиеся на несколько предыдущих значений
- Алгоритмы, опирающиеся на одно предыдущее значение

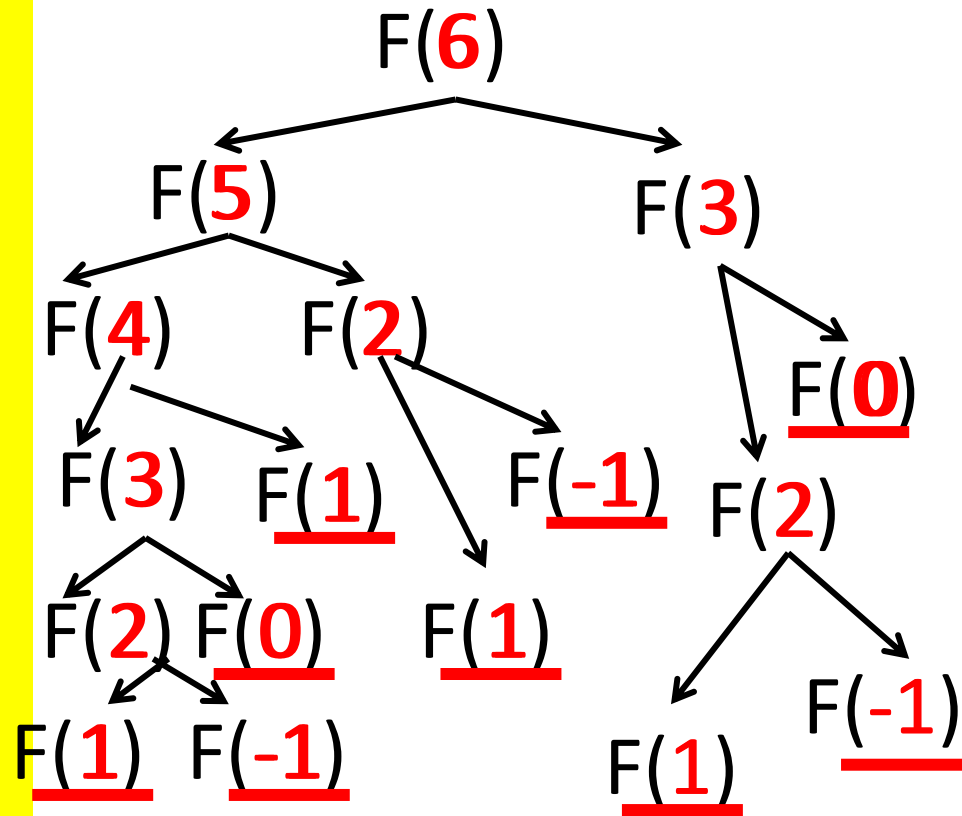
# Вызов рекурсивных процедур

№ 1. Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(6)?

Сумма = 28

Построение дерева вызовов

```
procedure F(n:integer);
begin
  writeln(n);
  if n > 1 then
    begin
      F(n - 1);
      F(n - 3)
    end
end;
```



Печать 6 5 4 3 2 1 -1 0 1 2 1 -1 3 2 1 -1 0

№ 2. Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(11)?

```
procedure F(n: integer); forward;
procedure G(n: integer); forward;
procedure F(n: integer);
begin
    if n > 0 then G(n - 1);
end;
procedure G(n: integer);
begin
    writeln('*');
    if n > 1 then F(n - 2);
end;
```

**построение  
последовательности  
вызовов**

F(11)

G(10) \*

F(8)

G(7) \*

F(5)

G(4) \*

F(2)

G(1) \*

**Ответ: 4**



Используя **Forward-описания** (предописания), вы можете делать процедуры или функции известными без фактического определения ее операторной части.

С точки предописания, другие процедуры и функции могут вызывать предописанную подпрограмму, делая возможной взаимную рекурсию.

№ 3. Чему будет равно значение, вычисленное при выполнении вызова  $F(7)$ ?

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else F := 1;
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n - 2)
  else G := 1;
end;
```

$$F(1) = 1$$

$$F(2) = 1$$

$$F(3) = F(2) + G(1) =$$

$$G(1) = 1$$

$$F(4) = F(3) + G(2) =$$

$$G(2) = 1$$

$$F(5) = F(4) + G(3) =$$

$$G(3) = G(2) + F(1) = 2$$

$$F(6) = F(5) + G(4) =$$

$$G(4) = G(3) + F(2) = 3$$

$$F(7) = F(6) + G(5) =$$

$$G(5) = G(4) + F(3) = 5$$

ОТВЕТ: 13

**Алгоритмы,  
опирающиеся на  
несколько предыдущих  
значений**

№ 4. Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(5)?

```
function F(n: integer): integer;  
begin  
    if n > 2 then F := F(n - 1) + F(n - 2)  
        else F := n;  
end;
```

$$\begin{aligned} F(5) &= \mathbf{F(4)} + \mathbf{F(3)} = \underline{\mathbf{F(3)}} + \underline{\mathbf{F(2)}} + \underline{\mathbf{F(2)}} + \underline{\mathbf{F(1)}} \\ &= \underline{\mathbf{F(2)} + \mathbf{F(1)}} + 2 + 2 + 1 = 2 + 1 + 2 + 2 + 1 \\ &= 8 \end{aligned}$$

**№ 5.** Последовательность чисел задается рекуррентным соотношением:

$$F(1) = 0$$

$$F(2) = 1$$

$$F(3) = 1$$

$F(n) = F(n-3) + F(n-2) + F(n-1)$ , при  $n > 3$ , где  $n$  – натуральное число.

Чему равно девятое число в последовательности?

*В ответе запишите только натуральное число.*

$$F(4) = F(1) + F(2) + F(3) = 2$$

$$F(5) = F(2) + F(3) + F(4) = 4$$

$$F(6) = F(3) + F(4) + F(5) = 7$$

$$F(7) = F(4) + F(5) + F(6) = 13$$

$$F(8) = F(5) + F(6) + F(7) = 24$$

$$F(9) = F(6) + F(7) + F(8) = 44$$

**Алгоритмы,  
опирающиеся на одно  
предыдущее значение**

**№ 6.** Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 3$$

$$F(n) = F(n-1) * (n-1), \text{ при } n > 1$$

Чему равно значение функции  $F(6)$ ?


$$F(2) = F(1) * 1 = 3$$

$$F(3) = F(2) * 2 = 6$$

$$F(4) = F(3) * 3 = 18$$

$$F(5) = F(4) * 4 = 72$$

$$F(6) = F(5) * 5 = 360$$



Используемые источники:

1. В.Р. Лещинер, М.А. Ройтберг. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ для учителей, подготовленные на основе анализа типичных ошибок участников ЕГЭ 2016 года по ИНФОРМАТИКЕ и ИКТ. – ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ, Москва, 2015
2. <http://inf.ege.ru>
3. Сайт «Решу ЕГЭ».

