

Тема 15. Реверсивный алгоритм (задание 11)

При решении задач данного вида необходимо знать:

- 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;
- рекурсия позволяет заменить цикл и в некоторых сложных задачах делает решение более понятным, хотя часто менее эффективным; существуют языки программирования, в которых рекурсия используется как один из основных приемов обработки данных (Lisp, Haskell).

Пример №1 с решением

Ниже записаны две рекурсивные процедуры: F и G:

```
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;
```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(11)?

Решение:

1) заметим, что каждая функция вызывает другую (это называется косвенная рекурсия), причём только один раз

вот цепочка вызовов:

$F(11) \rightarrow G(10) \rightarrow F(8) \rightarrow G(7) \rightarrow F(5) \rightarrow G(4) \rightarrow F(2) \rightarrow G(1)$

за один вызов функции G выводится одна звёздочка, внутри функции F звёздочки не выводятся, поэтому за 4 вызова G будет выведено 4 звёздочки

Ответ: **4.**

Пример №2 с решением

Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик	Python
SUB F(n) PRINT n IF n < 5 THEN	def F(n): print(n) if n < 5:

F(n + 1) F(n + 3) END IF END SUB	F(n + 1) F(n + 3)
Паскаль	Алгоритмический язык
<pre> procedure F(n: integer); begin writeln(n); if n < 5 then begin F(n + 1); F(n + 3) end end end </pre>	<pre> алг F(цел n) нач вывод n, нс если n < 5 то F(n + 1) F(n + 3) все кон </pre>
Си	
<pre> void F(int n) { printf("%d\n", n); if (n < 5) { F(n + 1); F(n + 3); } } </pre>	

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(1)?

Решение:

Первым действием процедура F(1) выведет число 1. Далее процедура F(1) вызовет процедуру F(n + 1), в результате выполнения которой на экране появится число $n + 1 = 2$. Процедура F(2) вызовет процедуру F(3), которая выведет на экран число 3 и вызовет процедуру F(4), которая выведет на экран число 4 и вызовет F(5), которая выведет на экран число 5.

После этого управление вернётся к процедуре F(4), которая начнёт выполнять следующий шаг своего алгоритма, т. е. обратиться к процедуре $F(n + 3) = F(7)$. Процедура F(7) выведет на экран число 7. Далее управление вернётся к процедуре F(3). Рассуждая аналогично приходим к выводу, что процедура F(3) дополнительно выведет на экран число 6, процедура F(2) — 5.

Последним действием процедуры F(1) будет вызов процедуры $F(n + 3) = F(4)$, которая выведет на экран числа 4, 5, 7.

Таким образом, на экране будут числа 1, 2, 3, 4, 5, 7, 6, 5, 4, 5, 7. Их сумма равна 49.

Ответ: **49**.

Пример №3 с решением

Дан рекурсивный алгоритм:

```

procedure F(n: integer);
begin
  writeln('*');
  if n > 0 then begin
    F(n-2);
    F(n div 2)
  end
end;

```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова F(7)?

Решение:

1) сначала определим рекуррентную формулу; обозначим через $G(n)$ количество звездочек, которые выводит программа при вызове $F(n)$

2) из программы видим, что

$$G(n) = 1 \text{ при всех } n \leq 0$$

$$G(n) = 1 + G(n-2) + G(n \div 2) \text{ при } n > 0$$

3) вспомним, что $n \div 2$ — это частное от деления n на 2

4) по этим формулам заполняем таблицу, начиная с нуля:

$$G(0) = 1$$

$$G(1) = 1 + G(-1) + G(0) = 1 + 1 + 1 = 3$$

$$G(2) = 1 + G(0) + G(1) = 1 + 1 + 3 = 5$$

$$G(3) = 1 + G(1) + G(1) = 1 + 3 + 3 = 7$$

$$G(4) = 1 + G(2) + G(2) = 1 + 5 + 5 = 11$$

$$G(5) = 1 + G(3) + G(2) = 1 + 7 + 5 = 13$$

$$G(6) = 1 + G(4) + G(3) = 1 + 11 + 7 = 19$$

$$G(7) = 1 + G(5) + G(3) = 1 + 13 + 7 = 21$$

n	0	1	2	3	4	5	6	7
G(n)	1	3	5	7	11	13	19	21

Ответ: **21**.

Задания для тренировки

1) Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1;$$

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

Чему равно значение функции $F(5)$? В ответе запишите только натуральное число.

2) Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(2) = 1$$

$$F(n) = F(n-1) * n - 2 * F(n-2), \text{ при } n > 2$$

Чему равно значение функции $F(6)$?

В ответе запишите только натуральное число.

3) Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) + 2^{n-1}, \text{ если } n > 1.$$

Чему равно значение функции $F(12)$?

В ответе запишите только натуральное число.

4) Ниже на пяти языках программирования записан рекурсивный алгоритм F .

Бейсик	Python
<pre>SUB F(n) PRINT n IF n < 5 THEN F(n + 1)</pre>	<pre>def F(n): print(n)</pre>

F(n + 2) END IF END SUB	if n < 5: F(n + 1) F(n + 2)
Алгоритмический язык	Паскаль
алг F(цел n) нач вывод n, нс если n < 5 то F(n + 1) F(n + 2) все кон	procedure F(n: integer); begin writeln(n); if n < 5 then begin F(n + 1); F(n + 2) end end end
Си	
void F(int n) { printf("%d\n", n); if (n < 5) { F(n + 1); F(n + 2); } } }	

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(2)?

5) Ниже на четырёх языках программирования записан рекурсивный алгоритм F.

Бейсик	Паскаль
SUB F(n) IF n > 0 THEN F(n - 4) PRINT n F(n \ 3) END IF END SUB	procedure F(n: integer); begin if n > 0 then begin F(n - 4); writeln(n); F(n div 3) end end; end;
Си	Алгоритмический язык
void F(int n) {	алг F(цел n) нач

<pre> if (n > 0) { F(n - 4); printf("%d\n", n); F(n / 3); } </pre>	<pre> если n > 0 то F(n - 4) вывод n, нс F(div(n, 3)) все кон </pre>
---	---

Ответы к заданиям для тренировки

1. 360
2. 44
3. 4095
4. 40
5. 20