

## 11 (базовый уровень, время – 5 мин)

Тема: рекурсивные алгоритмы.

Что нужно 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 задания:

**P-05.** Ниже записаны две рекурсивные процедуры:  $F$  и  $G$ :

```
def F(n):  
    if n > 0:  
        G(n - 1)  
def G(n):  
    print('*')  
    if n > 1:  
        F(n - 2)
```

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
- 2) вот цепочка вызовов:  
 $F(11) \rightarrow G(10) \rightarrow F(8) \rightarrow G(7) \rightarrow F(5) \rightarrow G(4) \rightarrow F(2) \rightarrow G(1)$
- 3) за один вызов функции  $G$  выводится одна звёздочка, внутри функции  $F$  звёздочки не выводятся, поэтому за 4 вызова  $G$  будет выведено 4 звёздочки
- 4) Ответ: **4**.

Ещё пример задания:

**P-05.** Дан рекурсивный алгоритм:

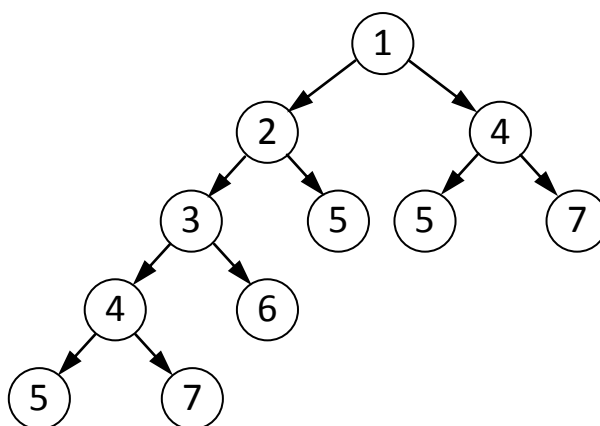
```
def F(n):  
    print(n)  
    if n < 5:  
        F(n + 1)  
        F(n + 3)
```

Найдите сумму чисел, которые будут выведены при вызове  $F(1)$ .

Решение (вариант 1, построение дерева вызовов):

- 1) поскольку в начале каждого вызова на экран выводится значение единственного параметра функции, достаточно определить порядок рекурсивных вызовов и сложить значения параметров

- 2) поскольку при  $n < 5$  выполняется два рекурсивных вызова, решать такую задачу «на бумажке» удобно в виде двоичного дерева (в узлах записаны значения параметров при вызове функции):



- 3) складывая все эти числа, получаем 49  
4) ответ: 49.

**Решение (вариант 2, подстановка):**

- 1) можно обойтись и без дерева, учитывая, что при каждом вызове с  $n < 5$  происходит два рекурсивных вызова; сумму чисел, полученных при вызове  $F(n)$ , обозначим через  $S(n)$  :

$$S(n) = \begin{cases} n + S(n+1) + S(n+3), & n < 5 \\ n, & n \geq 5 \end{cases}$$

- 2) выполняем вычисления:

$$S(1) = 1 + S(2) + S(4)$$

$$S(2) = 2 + S(3) + S(5) = 7 + S(3)$$

$$S(3) = 3 + S(4) + S(6) = 9 + S(4)$$

$$S(4) = 4 + S(5) + S(7) = 16$$

- 3) теперь остаётся вычислить ответ «обратным ходом»:

$$S(3) = 9 + 16 = 25$$

- 4)  $S(2) = 7 + 25 = 32$

$$S(1) = 1 + 32 + 16 = 49$$

- 5) Ответ: 49.

**Ещё пример задания:**

**P-04.** Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 6:  
        F(n+2)  
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове  $F(1)$ .

**Решение (вариант 1, метод подстановки):**

- 1) сначала определим рекуррентную формулу; обозначим через  $G(n)$  сумму чисел, которая выводится при вызове  $F(n)$   
2) при  $n \geq 6$  процедура выводит число  $n$  и заканчивает работу без рекурсивных вызовов:  
 $G(n) = n$  при  $n \geq 6$   
3) при  $n < 6$  процедура выводит число  $n$  и дважды вызывает сама себя:

$$G(n) = n + G(n+2) + G(3n) \text{ при } n < 6$$

- 4) в результате вызова  $F(1)$  получаем
 
$$G(1) = 1 + G(3) + G(3)$$

$$G(3) = 3 + G(5) + G(9) = 3 + G(5) + 9$$

$$G(5) = 5 + G(7) + G(15) = 5 + 7 + 15 = 27$$
- 5) используем обратную подстановку:
 
$$G(3) = 3 + G(5) + 9 = 3 + 27 + 9 = 39$$

$$G(1) = 1 + 2 * G(3) = 79$$
- 6) Ответ: **79**.

**Решение (вариант 2, динамическое программирование):**

- 1) п. 1-3 такие же, как в первом варианте решения
- 2) заполняем таблицу  $G(n)$  при  $n \geq 6$  (где  $G(n) = n$ )

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G(n)						6	7	8	9	10	11	12	13	14	15

- 3) остальные ячейки заполняем, начиная с  $n = 5$  справа налево, используя формулу:
 
$$G(n) = n + G(n+2) + G(3n)$$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G(n)	79	30	39	22	27	6	7	8	9	10	11	12	13	14	15

- 4) ответ читаем в самой левой ячейке
- 5) Ответ: **79**.

**Пример задания:**

**P-03.** Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        F(n-2)
        F(n // 2)
```

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

**Решение (вариант 1, составление полной таблицы):**

- 1) сначала определим рекуррентную формулу; обозначим через  $G(n)$  количество звездочек, которые выводит программа при вызове  $F(n)$
- 2) из программы видим, что
 
$$G(n) = 1 \text{ при всех } n \leq 0$$

$$G(n) = 1 + G(n-2) + G(n // 2) \text{ при } n > 0$$
- 3) вспомним, что  $n // 2$  – 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

- 5) Ответ: **21**.

**Решение (вариант 2, «с конца»):**

- 1) пп. 1-3 – как в варианте 1
- 2) по формулам  $G(7) = 1 + G(5) + G(3)$ , поэтому нужно найти  $G(5)$  и  $G(3)$
- 3)  $G(5) = 1 + G(3) + G(2)$ , нужны  $G(3)$  и  $G(2)$

- 4)  $G(3) = 1 + G(1) + G(1)$ , нужно  $G(1)$
- 5)  $G(2) = 1 + G(0) + G(1) = 2 + G(1)$ , нужно  $G(1)$
- 6)  $G(1) = 1 + G(-1) + G(0) = 1 + 1 + 1 = 3$
- 7) теперь идем «обратным ходом»:  
 $G(2) = 2 + G(1) = 5$   
 $G(3) = 1 + G(1) + G(1) = 1 + 3 + 3 = 7$   
 $G(5) = 1 + G(3) + G(2) = 1 + 7 + 5 = 13$   
 $G(7) = 1 + G(5) + G(3) = 1 + 13 + 7 = 21$
- 8) Ответ: **21**.

### Ещё пример задания:

**P-02.** Алгоритм вычисления значений функций  $F(n)$  и  $G(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = F(n-1) - G(n-1),$$

$$G(n) = F(n-1) + G(n-1), \text{ при } n \geq 2$$

Чему равно значение величины  $F(5)/G(5)$ ?

В ответе запишите только целое число.

#### Решение:

- 1) фактически рекуррентная формула задана для пары  $(F(n), G(n))$
- 2) замечаем, что  $F(n)$  – это разность предыдущей пары, а  $G(n)$  – сумма тех же значений
- 3) заполняем таблицу, начиная с известной первой пары

n	1	2	3	4	5
F(n)	1	0	-2	-4	-4
G(n)	1	2	2	0	-4

- 4) искомое значение  $F(5)/G(5)$  равно 1
- 5) ответ: **1**.

### Ещё пример задания:

**P-01.** Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

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

Чему равно значение функции  $F(5)$ ?

В ответе запишите только целое число.

#### Решение:

- 1) используя заданную рекуррентную формулу, находим, что  
 $F(5) = F(4) * 5$
- 2) применив формулу еще несколько раз, получаем  
 $F(5) = F(3) * 4 * 5 = F(2) * 3 * 4 * 5 = F(1) * 2 * 3 * 4 * 5$
- 3) мы дошли до базового случая, который останавливает рекурсию, так как определяет значение  $F(1) = 1$
- 4) окончательно  $F(5) = 1 * 2 * 3 * 4 * 5 = 120$
- 5) ответ: **120**.

### Ещё пример задания:

**P-00.** Процедура  $F(n)$ , где  $n$  – натуральное число, задана следующим образом (язык Паскаль):

```
def F(n):  
    if n < 3:  
        print('*')  
    else:
```

**F(n-1)**

**F(n-2)**

**F(n-2)**

Сколько звездочек напечатает эта процедура при вызове F(6)? В ответе запишите только целое число.

**Решение:**

- 1) эта задача по сути такая же, как и предыдущая, но «завёрнута» в другой фантик: для  $n < 3$  (то есть, для 1 и 2) функция выводит одну звездочку

$$F(1) = F(2) = 1$$

а для больших  $n$  имеем рекуррентную формулу

$$\begin{aligned} F(n) &= F(n-1) + F(n-2) + F(n-2) \\ &= F(n-1) + 2 * F(n-2) \end{aligned}$$

- 2) запишем в таблицу базовые случаи

n	1	2	3	4	5	6
F(n)	1	1				

- 3) заполняем таблицу, используя рекуррентную формулу:

n	1	2	3	4	5	6
F(n)	1	1	3	5	11	21

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

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

$$F(5) = F(4) + 2 * F(3) = 11$$

$$F(6) = F(5) + 2 * F(4) = 21$$

- 4) ответ: **21**.

## Задачи для тренировки<sup>1</sup>:

- 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 целое число.

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

$$F(1) = 1$$

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

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

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

$$F(1) = 1$$

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

Чему равно значение функции  $F(4)$ ? В ответе запишите только целое число.

- 4) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

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

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

- 5) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) * (3*n - 2), \text{ при } n > 1$$

Чему равно значение функции  $F(4)$ ? В ответе запишите только целое число.

- 6) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(0) = 1, F(1) = 1$$

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

Чему равно значение функции  $F(7)$ ? В ответе запишите только целое число.

- 7) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(0) = 1, F(1) = 1$$

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

Чему равно значение функции  $F(6)$ ? В ответе запишите только целое число.

- 8) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(0) = 1, F(1) = 1$$

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

Чему равно значение функции  $F(6)$ ? В ответе запишите только целое число.

- 9) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(0) = 1, F(1) = 1$$

$$F(n) = 3*F(n-1) - F(n-2), \text{ при } n > 1$$

<sup>1</sup> Источники заданий:

1. Демонстрационные варианты ЕГЭ 2013 гг.  
2. Проверочные работы МИОО.

Чему равно значение функции  $F(6)$ ? В ответе запишите только целое число.

- 10) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(0) = 1, F(1) = 1$$

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

Чему равно значение функции  $F(6)$ ? В ответе запишите только целое число.

- 11) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(0) = 1, F(1) = 1$$

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

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

- 12) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1, F(2) = 1$$

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

Чему равно значение функции  $F(7)$ ? В ответе запишите только целое число.

- 13) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1, F(2) = 1$$

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

Чему равно значение функции  $F(8)$ ? В ответе запишите только целое число.

- 14) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1, F(2) = 1$$

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

Чему равно значение функции  $F(7)$ ? В ответе запишите только целое число.

- 15) Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1, F(2) = 1$$

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

Чему равно значение функции  $F(8)$ ? В ответе запишите только целое число.

- 16) Алгоритм вычисления значения функции  $F(w)$ , где  $w$  - натуральное число, задан следующими соотношениями:

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

$$F(w) = 5 * F(w-1) - 4 * F(w-2) \text{ при } w > 2.$$

Чему равно значение функции  $F(15)$ ?

- 17) Алгоритм вычисления значения функции  $F(w)$ , где  $w$  - натуральное число, задан следующими соотношениями:

$$F(1) = 4; F(2) = 5;$$

$$F(w) = 4 * F(w-1) - 3 * F(w-2) \text{ при } w > 2.$$

Чему равно значение функции  $F(8)$ ?

- 18) (<http://ege.yandex.ru>) Алгоритм вычисления значений функций  $F(w)$  и  $Q(w)$ , где  $w$  - натуральное число, задан следующими соотношениями:

$$F(1) = 1; Q(1) = 1;$$

$$F(w) = F(w-1) + 2 * Q(w-1) \text{ при } w > 1$$

$$Q(w) = Q(w-1) - 2 * F(w-1) \text{ при } w > 1.$$

Чему равно значение функции  $F(5)+Q(5)$ ?

- 19) Алгоритм вычисления значения функции  $F(w)$ , где  $w$  - натуральное число, задан следующими соотношениями:

$$F(1) = 1; F(2) = 2;$$

$$F(w) = 3 * F(w-1) - 2 * F(w-2) \text{ при } w > 2.$$

Чему равно значение функции F(7)?

- 20) Алгоритм вычисления значения функции F(w), где w - натуральное число, задан следующими соотношениями:

$$F(1) = 2; F(2) = 4;$$

$$F(w) = 4 * F(w-1) - 3 * F(w-2) \text{ при } w > 2.$$

Чему равно значение функции F(7)?

- 21) (<http://ege.yandex.ru>) Алгоритм вычисления значения функции F(n), где n - натуральное число, задан следующими соотношениями:

$$F(1) = 1; F(2) = 2;$$

$$F(n) = 5 * F(n-1) - 6 * F(n-2) \text{ при } n > 2.$$

Чему равно значение функции F(7)?

- 22) (<http://ege.yandex.ru>) Алгоритм вычисления значения функции F(n), где n - натуральное число, задан следующими соотношениями:

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

$$F(n) = F(n-3) * (n-1) / 3 \text{ при } n > 3.$$

Чему равно значение функции F(16)?

- 23) Алгоритм вычисления значений функций F(n) и G(n), где n – натуральное число, задан следующими соотношениями:

$$F(1) = 2; G(1) = 1;$$

$$F(n) = F(n-1) - G(n-1),$$

$$G(n) = F(n-1) + G(n-1), \text{ при } n \geq 2$$

Чему равно значение величины F(5)/G(5)? В ответе запишите только целое число.

- 24) Алгоритм вычисления значений функций F(n) и G(n), где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = F(n-1) - G(n-1),$$

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

Чему равно значение величины F(5)/G(5)? В ответе запишите только целое число.

- 25) Алгоритм вычисления значений функций F(n) и G(n), где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = F(n-1) - 2 * G(n-1),$$

$$G(n) = F(n-1) + G(n-1), \text{ при } n \geq 2$$

Чему равно значение величины G(5)/F(5)? В ответе запишите только целое число.

- 26) Алгоритм вычисления значений функций F(n) и G(n), где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = 2 * F(n-1) - G(n-1),$$

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

Чему равно значение величины G(5)+F(5)? В ответе запишите только целое число.

- 27) Алгоритм вычисления значений функций F(n) и G(n), где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = 2 * F(n-1) - G(n-1),$$

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

Чему равно значение величины F(5)-G(5)? В ответе запишите только целое число.

- 28) Алгоритм вычисления значений функций F(n) и G(n), где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$



$$F(n) = F(n-1) - 2*G(n-1),$$

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

Чему равно значение величины  $G(5)-F(5)$ ? В ответе запишите только целое число.

- 29) Алгоритм вычисления значений функций  $F(n)$  и  $G(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = 3*F(n-1) - 2*G(n-1),$$

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

Чему равно значение величины  $G(5)-F(5)$ ? В ответе запишите только целое число.

- 30) Алгоритм вычисления значений функций  $F(n)$  и  $G(n)$ , где  $n$  – натуральное число, задан следующими соотношениями:

$$F(1) = 1; G(1) = 1;$$

$$F(n) = 3*F(n-1) - 3*G(n-1),$$

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

Чему равно значение величины  $F(5)-G(5)$ ? В ответе запишите только целое число.

- 31) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        F(n-2)
        F(n // 2)
        F(n // 2)
```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова  $F(5)$ ?

- 32) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        F(n-2)
        F(n-2)
        F(n // 2)
```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова  $F(6)$ ?

- 33) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        F(n-3)
        F(n // 2)
```

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

- 34) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        F(n-3)
        F(n-2)
        F(n // 2)
```

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

- 35) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        F(n-3)
```

```
F(n-2)
F(n // 2)
F(n // 2)
```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова F(6)?

36) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        print('*')
        F(n-2)
        F(n // 2)
```

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

37) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        print('*')
        F(n-2);
        F(n // 2)
        F(n // 2)
```

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

38) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 0:
        print('*')
        F(n-2)
        F(n-2)
        F(n // 2)
```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова F(6)?

39) Дан рекурсивный алгоритм:

```
def F(n):
    if n > 0:
        F(n-2)
        F(n-1)
        F(n-1)
    print('*')
```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова F(5)?

40) Дан рекурсивный алгоритм:

```
def F(n):
    if n > 0:
        print('*')
        F(n-2)
        F(n-1)
        F(n-1)
    print('*')
```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова F(5)?

41) Дан рекурсивный алгоритм:

```
def F(n):
    if n > 1:
        F(n-2)
```

```
F(n-1)
F(n // 2)
print('*')
```

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

42) Дан рекурсивный алгоритм:

```
def F(n):
    if n > 2:
        print('*')
        F(n-2)
        F(n-1)
        F(n // 2)
    print('*')
```

Сколько символов "звездочка" будет напечатано на экране при выполнении вызова F(6)?

43) Алгоритм вычисления значения функции F(n), где n – натуральное число, задан следующими соотношениями:

$F(1) = 1,$   
 $F(n) = F(n-1) + 2^{n-1},$  при  $n > 1$

Чему равно значение функции F(12)? В ответе запишите только целое число.

44) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n < 6:
        F(n+2)
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(2).

45) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n < 5:
        F(n+2)
        F(n*2)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

46) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n < 5:
        F(n+3)
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

47) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n < 7:
        F(n+3)
        F(n*2)
```

Найдите сумму чисел, которые будут выведены при вызове F(2).

48) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n < 7:
        F(n+2)
```

**F(n+3)**

Найдите сумму чисел, которые будут выведены при вызове F(1).

49) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 5:  
        F(n+2)  
        F(n+3)  
        F(n*2)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

50) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 5:  
        F(n+1)  
        F(n+2)  
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(2).

51) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 6:  
        print(n)  
        F(n+2)  
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(2).

52) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 5:  
        print(n)  
        F(n+3)  
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

53) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 6:  
        print(n)  
        F(n+2)  
        F(n+3)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

54) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 7:  
        print(n)  
        F(n+1)  
        F(n+2)  
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(2).

55) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 6:  
        print(n)  
        F(n+1)  
        F(n+2)  
        F(n*2)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

56) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 6:  
        print(n)  
        F(n+1)  
        F(n*2)  
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(2).

57) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n < 7:  
        print(n)  
        F(n+2)  
        F(n*2)  
        F(n*3)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

58) Алгоритм вычисления значения функции F(n), где n – натуральное число, задан следующими соотношениями:

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

Чему равно значение функции F(8)?

59) Алгоритм вычисления значения функции F(n), где n – натуральное число, задан следующими соотношениями:

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

Чему равно значение функции F(7)?

60) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n > 0:  
        F(n-1)  
        F(n-3)
```

Найдите сумму чисел, которые будут выведены при вызове F(5).

61) Дан рекурсивный алгоритм:

```
def F(n):  
    print(n)  
    if n > 1:  
        F(n-3)  
        F(n-1)
```

Найдите сумму чисел, которые будут выведены при вызове F(6).

62) Дан рекурсивный алгоритм:

```
def F(n):  
    if n > 2:  
        return F(n - 1) + F(n - 2)  
    else:  
        return n
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(5)?

63) (И. Тощенко) Дан рекурсивный алгоритм:

```
def F(n):  
    if n > 3:  
        return F(n - 1) * F(n - 2)  
    else:  
        return n
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(6)?

64) (И. Тощенко) Дан рекурсивный алгоритм:

```
def F(n):  
    if n >= 3:  
        return F(n-3) + F(n-2)*F(n-1)  
    else:  
        return n
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(7)?

65) (И. Тощенко) Дан рекурсивный алгоритм:

```
def F(n):  
    if n < 5:  
        return F(n+2) + F(n+3) + F(n+1)  
    else:  
        return n
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(2)?

66) (И. Тощенко) Дан рекурсивный алгоритм:

```
def F(n):  
    if n < 5:  
        return F(n*3) + F(n+3) + F(n+1)  
    else:  
        return n // 2
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(2)?

67) (И. Тощенко) Дан рекурсивный алгоритм:

```
def F(n):  
    if n < 5:  
        return F(n+3) + F(2*n) + F(3*n // 2)  
    else:  
        return n + 2
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(3)?

68) (И. Тощенко) Дан рекурсивный алгоритм:

```
def F(n):  
    if n < 6:  
        return n+F(n+3) * F(2*n)  
    else:  
        return n*2
```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(3)?

69) (И. Тощенко) Дан рекурсивный алгоритм:

```
def F(n):
```

```

if n > 1:
    return 2*n + F(n-3) + F(n-2)
else:
    return n + 5

```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(6)?

70) Ниже записаны две рекурсивные процедуры, F и G:

```

def F(n):
    if n > 0:
        G(n - 1)

def G(n):
    print('*')
    if n > 1:
        print('*')
        F(n - 2)

```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(13)?

71) Ниже записаны две рекурсивные процедуры, F и G:

```

def F(n):
    print('*')
    if n > 0:
        G(n - 1)

def G(n):
    print('*')
    if n > 1:
        F(n - 2)

```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(13)?

72) Ниже записаны две рекурсивные процедуры, F и G:

```

def F(n):
    print('*')
    if n > 0:
        print('*')
        G(n - 1)

def G(n):
    print('*')
    if n > 1:
        F(n - 2)

```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(12)?

73) Ниже записаны две рекурсивные процедуры, F и G:

```

def F(n):
    print('*')
    if n > 0:
        print('*')
        G(n - 1)

def G(n):
    print('*')

```

```

if n > 1:
    print('*')
    F(n - 2)

```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(12)?

74) Ниже записан рекурсивный алгоритм F:

```

def F(n):
    if n > 2:
        return F(n-1)+F(n-2)+F(n-3)
    else:
        return n

```

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(6)?

75) Ниже записаны две рекурсивные процедуры, F и G:

```

def F(n):
    if n > 0:
        return G(n - 1)
def G(n):
    print('*')
    if n > 1:
        F(n - 3)

```

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

76) Ниже записаны две рекурсивные функции, F и G:

```

def F(n):
    if n > 2:
        return F(n - 1) + G(n - 2)
    else:
        return 1
def G(n):
    if n > 2:
        return G(n - 1) + F(n - 2)
    else:
        return 1

```

Чему будет равно значение, вычисленное при выполнении вызова F(7)?

77) Ниже записаны две рекурсивные функции, F и G:

```

def F(n):
    if n > 2:
        return F(n - 1) + G(n - 2)
    else:
        return n
def G(n):
    if n > 2:
        return G(n - 1) + F(n - 2)
    else:
        return n+1

```

Чему будет равно значение, вычисленное при выполнении вызова F(6)?

78) (А.Г. Гильдин, Уфа). Вызов представленной ниже рекурсивной функции приводит к появлению на экране чисел и точек. С каким минимальным натуральным аргументом, а нужно вызвать эту функцию, чтобы в результате на экране появилось 5 точек (не обязательно подряд, между точками могут встречаться числа)?

```

def gz(a):
    if a < 1:

```



```
    return 1
if a % 3 == 0:
    print('...')
    p = gz(a // 3) + gz(a // 4)
else:
    print('.')
    p = gz(a // 4)
print(p)
return 2
```

79) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n > 1:
        print(n)
        F(n - 2)
        F(n - 5)
```

Найдите сумму чисел, которые будут выведены при вызове F(9).

80) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n > 1:
        print(n)
        F(n - 1)
        F(n - 4)
```

Найдите сумму чисел, которые будут выведены при вызове F(7).

81) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n > 1:
        print(n)
        F(n - 2)
        F(n - 3)
```

Найдите сумму чисел, которые будут выведены при вызове F(6).

82) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n > 1:
        print(n)
        F(n - 1)
        F(n - 3)
```

Найдите сумму чисел, которые будут выведены при вызове F(5).

83) (С.С. Юнусова) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    if n < 6:
        n = n + 1
        F(n + 1)
        F(n * 2)
    print(n)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

84) (С.С. Юнусова) Дан рекурсивный алгоритм:

```
def F(n):
    print(n)
    n = n + 1
    if n < 7:
        print(n)
        F(n + 1)
        F(n + 2)
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

85) (С.С. Юнусова) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n > 5:
        F(n - 1)
        F(n - 3)
        print('*')
    else:
        if n > 3:
            F(n // 2)
            print('*')
```

Найдите количество звездочек, которые будут выведены при вызове F(8).

86) (С.С. Юнусова) Дан рекурсивный алгоритм:

```
def F(n):
    print('*')
    if n < 5:
        F(n + 3)
        print('*')
    if n <= 3:
        F(n + 2)
        print('*')
```

Найдите количество звездочек, которые будут выведены при вызове F(1).

87) Даны две рекурсивные функции:

```
def F(n):
    if n > 1:
        return F(n-1)+ G(n-1)
    else: return n
def G(n):
    if n > 1:
        return G(n-1)+ F(n)
    else: return n
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

88) Даны две рекурсивные функции:

```
def F(n):
    if n > 1:
        return F(n-1)+ G(n-1)
    else: return n-1
def G(n):
    if n > 1:
        return G(n-1)+ F(n)
    else: return n+1
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

89) Даны две рекурсивные функции:

```
def F(n):  
    if n > 1:  
        return F(n-1)+ G(n-1)  
    else: return n+1  
def G(n):  
    if n > 1:  
        return G(n-1)+ F(n)  
    else: return n-1
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

90) Даны две рекурсивные функции:

```
def F(n):  
    if n > 1:  
        return F(n-1)+ G(n-1)  
    else: return n+1  
def G(n):  
    if n > 1:  
        return G(n-1)+ F(n)  
    else: return 2*n
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

91) Даны две рекурсивные функции:

```
def F(n):  
    if n > 1:  
        return F(n-1)+ G(n-1)  
    else: return 2*n  
def G(n):  
    if n > 1:  
        return G(n-1)+ F(n)  
    else: return n-2
```

Чему будет равно значение, вычисленное при выполнении вызова F(5)?

92) Даны две рекурсивные функции:

```
def F(n):  
    if n > 2:  
        return F(n-1)+ G(n-2)  
    else: return n  
def G(n):  
    if n > 2:  
        return G(n-1)+ F(n-2)  
    else: return 2-n
```

Чему будет равно значение, вычисленное при выполнении вызова G(6)?

93) Даны две рекурсивные функции:

```
def F(n):  
    if n > 2:  
        return F(n-1)+ G(n-2)  
    else: return n+2  
def G(n):  
    if n > 2:  
        return G(n-1)+ F(n-2)  
    else: return 3-n
```

Чему будет равно значение, вычисленное при выполнении вызова G(6)?

94) Даны две рекурсивные функции:

```
def F(n):
```

```
if n > 2:
    return F(n-1)+ G(n-2)
else: return 2
def G(n):
    if n > 2:
        return G(n-1)+ F(n-2)
    else: return 3
```

Чему будет равно значение, вычисленное при выполнении вызова G(6)?

95) Даны две рекурсивные функции:

```
def F(n):
    if n > 2:
        return F(n - 1)+ G(n - 2)
    else: return n
def G(n):
    if n > 2:
        return G(n - 1)+ F(n - 2)
    else: return n+1
```

Чему будет равно значение, вычисленное при выполнении вызова F(6)?

96) Даны две рекурсивные функции:

```
def F(n):
    if n > 2:
        return F(n - 1)+ G(n - 2)
    else: return n-1
def G(n):
    if n > 2:
        return G(n - 1)+ F(n - 2)
    else: return n+1
```

Чему будет равно значение, вычисленное при выполнении вызова G(7)?

97) (Е. Филина-Поликарпова) Что выведет программа при вызове F(5)?

```
def F(n):
    print(n, end='')
    if n >= 3:
        F(n - 1)
        F(n - 3)
```

98) (Е. Филина-Поликарпова) Что выведет программа при вызове F(6)?

```
def F(n):
    print(n, end='')
    if n >= 3:
        F(n - 1)
        F(n - 3)
```

99) (Е. Филина-Поликарпова) Что выведет программа при вызове F(5)?

```
def F(n):
    print(n, end='')
    if n >= 3:
        F(n - 1)
        F(n - 2)
```

100) (Е. Филина-Поликарпова) Что выведет программа при вызове F(5)?

```
def F(n):
    print(n+1, end='')
    if n >= 3:
        F(n - 3)
```

F(n - 2)

101) (Е. Филина-Поликарпова) Что выведет программа при вызове F(5)?

```
def F(n):  
    if n >= 3:  
        print(n+1, end='')  
        F(n - 1)  
        F(n - 2)
```

102) (Е. Филина-Поликарпова) Что выведет программа при вызове F(7)?

```
def F(n):  
    if n >= 3:  
        print(n, end='')  
        F(n - 3)  
        F(n - 2)
```

103) (Е. Филина-Поликарпова) Что выведет программа при вызове F(8)?

```
def F(n):  
    if n > 3:  
        print(n, end='')  
        F(n - 3)  
        F(n - 2)
```

104) (Е. Филина-Поликарпова) Что выведет программа при вызове F(-1)?

```
def F(n):  
    if n < 3:  
        print(n, end='')  
        F(n + 3)  
        F(n + 2)
```

105) (А.Н. Носкин) Сколько символов 'A' выведет программа при вызове F(10)?

```
def F ( n ):  
    if n > 0:  
        print('B', end='')  
        G(n - 1)  
def G ( n ):  
    if n > 1:  
        print('A', end='')  
        F(n - 2)
```

106) (А.Н. Носкин) Что выведет программа при вызове F(8)?

```
def F ( n ):  
    if n > 0:  
        print('B', end='')  
        G(n - 1)  
def G ( n ):  
    print('A', end='')  
    if n > 1:  
        print('A', end='')  
        F(n - 2)
```

107) (А.Н. Носкин) Что выведет программа при вызове F(5)?

```
def F ( n ):  
    print('A', end='')  
    if n > 0:  
        print('B', end='')  
        G(n - 1)  
def G ( n ):  
    print('C', end='')
```

```
if n > 1:  
    F(n - 2)
```

- 108) (А.Н. Носкин) При выполнении вызова F(8) на экран было выведено математическое выражение. Вычислите его значение.

```
def F( n ):  
    print("2", end="")  
    if n > 0:  
        print("*", end="")  
        G(n - 1)  
def G( n ):  
    print("3", end="")  
    if n > 1:  
        F(n - 2)
```

- 109) (А.Н. Носкин) При выполнении приведённой ниже программы на экран было выведено математическое выражение. Вычислите его значение.

```
def F( n ):  
    print("5", end="")  
    if n > 0:  
        print("+", end="")  
        G(n - 1)  
def G( n ):  
    print("2", end="")  
    if n > 1:  
        F(n - 2)  
print("2*(", end="")  
F(8)  
print(")", end="")
```

- 110) (А.Н. Носкин) Определите сумму цифр, которые выведет программа при вызове F(60)?

```
def F( n ):  
    if n > 0:  
        print("3", end="")  
        G(n - 1)  
def G( n ):  
    print("2", end="")  
    if n > 1:  
        print("2", end="")  
        F(n - 2)
```

- 111) (А.Н. Носкин) Определите сумму цифр, которые выведет программа при вызове F(90)?

```
def F( n ):  
    print("5", end="")  
    if n > 0:  
        print("2", end="")  
        G(n - 1)  
def G( n ):  
    if n > 1:  
        print("3", end="")  
        F(n - 2)
```

- 112) Задана функция F(n). Определите значение F(11).

```
def F(n):  
    if n > 2:  
        return F(n-2) + F(n//2)  
    else:
```

```
return n
```

113) Задана функция F(n). Определите значение F(15).

```
def F(n):  
    if n > 3:  
        return F(n-3) + F(n//3)  
    else:  
        return n
```

114) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):  
    if n > 0:  
        F(n - 4)  
        F(n // 3)  
        print(n, end="")
```

115) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):  
    if n > 0:  
        F(n - 4)  
        F(n // 2)  
        print(n, end="")
```

116) (Л.А. Тумарина, г. Электросталь) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):  
    if n > 0:  
        F(n - 4)  
        print(n, end="")  
        F(n // 2)
```

117) (Л.А. Тумарина, г. Электросталь) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):  
    if n > 3:  
        F(n - 4)  
        F(n // 2)  
        print(n, end="")
```

118) (Л.А. Тумарина, г. Электросталь) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):  
    print(n, end="")  
    if n > 3:  
        F(n - 4)  
        F(n // 2)
```

119) Определите, что выведет на экран программа при вызове F(5).

```
def F(n):  
    if n > 0:  
        F(n - 1)  
        print(n)  
        F(n // 4)
```

120) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):  
    if n > 0:  
        print(n)  
        F(n - 4)  
        F(n // 2)
```

121) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):
```

```
if n > 0:
    print(n)
    F(n // 2)
    F(n - 4)
```

122) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):
    if n > 0:
        F(n - 3)
        print(n)
        F(n // 3)
```

123) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):
    if n > 0:
        F(n // 3)
        print(n, end=" ")
        F(n - 3)
```

124) (О.Л. Дуркин) Определите, что выведет на экран программа при вызове F(4).

```
def f(n):
    if n > 0:
        f(n // 4)
        print(1, end=" ")
        f(n-3)
    print(2, end=" ")
```

125) (О.Л. Дуркин) Определите, что выведет на экран программа при вызове F(6).

```
def f(n):
    print(1, end=" ")
    if 1 < n:
        print(2, end=" ")
        f(n // 4)
        f(n-3)
```

126) (О.Л. Дуркин) Определите, что выведет на экран программа при вызове F(6).

```
def f(n):
    print(n+3, end=" ")
    if 1 < n:
        f(n // 4)
        f(n - 3)
        print(n, end=" ")
```

127) (О.Л. Дуркин) Определите, что выведет на экран программа при вызове F(0, 6).

```
def f(k, n):
    k += 1
    if k % 2 == 0 and n > 0:
        f(k, n // 3)
        print(n, end=" ")
        f(k, n-k)
    if k % 2 == 1 and n > 0:
        f(k, n//2)
        f(k, n-k-1)
        print(n, end=" ")
```

128) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(7).

```
def F(n):
    n += 1
    if n > 3:
```



```
    print(n, end='')
    F(n - 3)
    F(n // 3)
else:
    print(n, end='')
```

129) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):
    if n > 3:
        print(n, end='')
        n += 1
        F(n - 3)
        F(n // 3)
    else:
        print(n, end='')
```

130) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):
    if n > 3:
        print(n, end='')
        F(n - 3)
        n += 1
        F(n // 3)
    else:
        print(n, end='')
```

131) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(7).

```
def F(n):
    n += 1
    if n > 3:
        print(n, end='')
        F(n // 3)
        F(n - 3)
    else:
        print(n, end='')
```

132) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):
    if n > 3:
        print(n, end='')
        n += 1
        F(n // 3)
        F(n - 3)
    else:
        print(n, end='')
```

133) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(9).

```
def F(n):
    if n > 3:
        print(n, end='')
        F(n // 3)
        n += 1
        F(n - 3)
    else:
        print(n, end='')
```

134) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(4).

```
def F(n):
```

```

if n > 2:
    print(n, end='')
    F(n - 1)
    G(n - 2)
else:
    print(n+2, end='')

def G(n):
    print(n, end='')
    if n > 2:
        G(n - 1)
        F(n - 2)
    else:
        print(n+1, end='')

```

135) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(5).

```

def F(n):
    if n > 2:
        print(n, end='')
        F(n - 1)
        G(n - 2)
    else:
        print(n+2, end='')

def G(n):
    print(n, end='')
    if n > 2:
        G(n - 1)
        F(n - 2)

```

136) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(7).

```

def F(n):
    n -= 1
    if n > 2:
        print(n, end='')
        F(n - 1)
        G(n - 2)
    else:
        print(n+2, end='')

def G(n):
    print(n, end='')
    if n > 2:
        n -= 1
        G(n - 1)
        F(n - 2)

```

137) (Л.А. Тумарина) Определите, что выведет на экран программа при вызове F(6).

```

def F(n):
    n -= 1
    if n > 2:
        print(n, end='')
        F(n - 1)
        G(n - 2)
    else:

```

```

        print(n+2, end='')

def G(n):
    print(n, end='')
    if n > 2:
        n -= 1
        G(n - 1)
        F(n - 2)
    else:
        print(n+1, end='')

```

138) (Б.С. Мижлин) Определите, что выведет на экран программа.

```

def F(n):
    print('f',end='')
    if n>4:
        return F(n-2) + G(n-1)
    else:
        return 2*n
def G(n):
    if n>3:
        print('g',end='')
        return G(n-3) + F(n-5)
    else:
        return 2*(n+1)

a=7
print(F(a),end='')
print(a//2)

```

139) (Б.С. Мижлин) Определите, что выведет на экран программа.

```

def F(n):
    global s, sum
    if n>2:
        G(n-1)
        s+='f'*(n//2)
    sum+=n
def G(n):
    global s, sum
    if n>3:
        F(n-3)
        s+='g'*(n-2)
    sum+=n

s = 'f'
sum = 0
F(6)
print(str(sum)+s)

```