

Программное решение 16ых задач через рекурсию

[Курс подготовки к ЕГЭ по информатике 2027](#)

1	<u>ФИПИ 097227</u>	2
1.1	<u>Решение</u>	2
2	<u>Демоверсия 2023</u>	3
2.1	<u>Решение</u>	3
3	<u>Авторское задание</u>	4
3.1	<u>Решение</u>	4
4	<u>ФИПИ FDDA65</u>	5
4.1	<u>Решение</u>	5

ФИПИ 097227

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

$$F(n) = 1 \text{ при } n < 3;$$

$$F(n) = F(n-2) - F(n-1), \text{ если } n > 2 \text{ и при этом } n \text{ чётно};$$

$$F(n) = 2 \cdot F(n-1) - F(n-2), \text{ если } n > 2 \text{ и при этом } n \text{ нечётно}.$$

Чему равно значение функции $F(19)$?

Решение

Всё решение сводится к созданию рекурсивной функции и выводу нужного результата на экран.

```
1  def f(n):
2  if(n<3):
3      return 1
4  if(n>2) and (n%2==0):
5      return f(n-2)-f(n-1)
6  if(n>2) and (n%2!=0):
7      return 2*f(n-1)-f(n-2)
8  print(f(19))
```

Демоверсия 2023

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

$$F(n) = 1 \text{ при } n = 1;$$

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

Чему равно значение выражения $F(2023)/F(2020)$?

Решение

В данной задаче, если написать рекурсивную функцию, мы получим ошибку, связанную с максимально допустимой глубиной рекурсии. В python мы имеем возможность увеличить глубину рекурсии с помощью библиотеки sys. В данном примере будет достаточно установить максимальную глубину в 2000.

```
1  from sys import *
2  setrecursionlimit(2000)
3  def f(n):
4      if(n==1):
5          return 1
6      if(n>1):
7          return n*f(n-1)
8  print(f(2023)/f(2020))
```

Авторское задание

Алгоритм вычисления значения функции $F(n)$, где n - целое неотрицательное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n \leq 1;$$

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

Чему равно значение выражения $F(200)/F(199)$?

Решение

В этой задаче мы получим проблему со слишком долгим выполнением программы. Это связано с тем, что мы постоянно пересчитываем одни и те же значения, так как вызывается $F(n-1)$ и $F(n-2)$ для каждого заинтересовавшего нас значения n . Здесь нам на помощь приходит кэширование. Мы можем ввести некоторый список, в котором будут храниться все интересующие нас значения, и рассчитывать новые мы будем только в том случае, если этого значения еще нет в этом самом кэше, в противном же случае мы просто будем возвращать это значение из кэша.

Мы можем уже рассчитанные значения сохранять в некоторый список, и когда нам снова понадобится это значение, не рассчитывать его, а брать из этого списка.

```
1  cache=[None]*201
2  def f(n):
3      if cache[n]==None:
4          if n<=1:
5              cache[n]=1
6          if n>1:
7              cache[n]=f(n-1)+2*f(n-2)
8      return cache[n]
9  print(f(200)/f(199))
```

ФИПИ FDDA65

Алгоритм вычисления функций $F(n)$ и $G(n)$, где n - целое число, задан следующими соотношениями:

$$F(n) = 2 \times (G(n - 3) + 8);$$

$$G(n) = 2 \times n, \text{ если } n < 10;$$

$$G(n) = G(n - 2) + 1, \text{ если } n \geq 10.$$

Чему равно значение выражения $F(15548)$?

Решение

С двумя функциями принцип работы тот же, что и с одной. Просто теперь мы опишем принцип работы обеих функций.

```
1  from sys import *
2  setrecursionlimit(20000)
3
4  def f(n):
5      return 2*(g(n-3)+8)
6
7  def g(n):
8      if n<10:
9          return 2*n
10     if n>=10:
11         return g(n-2)+1
12
13  print(f(15548))
```
