

```
1
2 def F(n):
3     if n < 6:
4         return n
5     if n > 5 and n % 2 == 0:
6         return n + F(n//2) * 2
7     if n > 5 and n % 2 == 1:
8         return F(n-2) + F(n-1)
9
10 def digit_sum(n):
11     total = 0
12     for i in str(n):
13         total += int(i)
14     return total
15
16 for i in range(1, 1001):
17     if digit_sum(F(i)) == 22:
18         print(i)
19         break
```

Косяк был тут. Лучше использовать тут `//`, потому что в нашем решении есть функция связанная со строкой. А при обычном делении (`/`) у нас результат всегда float т.е. с нулем в конце 1234.0

И мы делаем 1234.0 строкой (`str(n)`), проходимся по значениям и каждый символ добавляем в `total`, но перед тем как добавить преобразовываем каждый символ в `int`, но возникает проблема когда доходим до точки. Программа не может точку преобразовать в число.

А если мы бы использовали целочисленное деление, то было бы просто 1234, что легко преобразуется в строку.

**Поэтому я переобулся, лучше везде пиши `//`, а не `/`**

Либо пиши два варианта, если ответы совпадут, то все ок, если где то ошибка то значит вариант выбора очевиден, либо если ответы разные то нужно уже аналитически подумать какое решение использовать, какое что нам дает