



ЕГЭ ПО ИНФОРМАТИКЕ С ЕВГЕНИЕМ ДЖОБСОМ

Дана программа для исполнителя Редактор:

НАЧАЛО

ПОКА нашлось (5555) ИЛИ нашлось (3333)

ЕСЛИ нашлось (5555)

ТО заменить (5555, 3)

ИНАЧЕ заменить (3333, 5)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Какая строка получится в результате применения приведённой выше программы к строке, состоящей из 147 идущих подряд цифр 3?





Перенесём алгоритм из задания в код на Python, прежде создав строку, состоящую из 147 троек:

```
s = '3'*147
while '5555' in s or '3333' in s:
    if '5555' in s:
        s = s.replace('5555', '3', 1)
    else:
        s = s.replace('3333', '5', 1)
print(s)
```

Получаем ответ: 555.





ЕГЭ ПО ИНФОРМАТИКЕ С ЕВГЕНИЕМ ДЖОБСОМ

Дана программа для исполнителя Редактор:

```
ПОКА нашлось (555) ИЛИ нашлось (888)
  ЗАМЕНИТЬ (555, 8)
  ЗАМЕНИТЬ (888, 55)
КОНЕЦ ПОКА
```

Известно, что начальная строка состоит более чем из 100 цифр 5 и не содержит других символов. В ходе работы алгоритма получилась строка, не содержащая цифр 8. Укажите минимальную возможную длину входной строки.





ЕГЭ ПО ИНФОРМАТИКЕ С ЕВГЕНИЕМ ДЖОБСОМ

Будем генерировать строку, состоящую из n (где $n > 100$) цифр 5. И найдём минимальное значение n , при котором в результате работы алгоритма получится строка, не содержащая восьмёрок:

```
for n in range(101, 1000):  
    s = '5'*n  
    while '555' in s or '888' in s:  
        s = s.replace('555', '8', 1)  
        s = s.replace('888', '55', 1)  
    if '8' not in s:  
        print(n)  
        break
```

Получаем ответ: 107.





ЕГЭ ПО ИНФОРМАТИКЕ С ЕВГЕНИЕМ ДЖОБСОМ

Какое количество вариантов строки можно получить в результате работы приведенного ниже алгоритма, если использовать четыре 1, три 3 и две 2 на входе?

НАЧАЛО

ПОКА нашлось (133) ИЛИ нашлось (221) ИЛИ нашлось (23)

ЕСЛИ нашлось (133)

ТО заменить (133, 1)

ЕСЛИ нашлось (221)

ТО заменить (221, 31)

ЕСЛИ нашлось (23)

ТО заменить (23, 21)

КОНЕЦ ПОКА

КОНЕЦ





ЕГЭ ПО ИНФОРМАТИКЕ С ЕВГЕНИЕМ ДЖОБСОМ

Нужно рассмотреть различные варианты входной строки, состоящей из 4 единиц, 3 троек и 2 двоек. Для этого можно воспользоваться модулем `itertools` и перебрать все перестановки этого набора цифр с помощью `permutations`:

```
from itertools import permutations

ss = [''.join(s) for s in\
      permutations('1'*4+'3'*3+'2'*2)]
```

Нас интересует количество различных результатов работы алгоритма, потому будем сохранять их во множество. И получим результат работы алгоритма для каждой из получившихся в списке `ss` строк ->





В результате имеем следующий код:

```
from itertools import permutations

ss = [''.join(s) for s in\
      permutations('1'*4+'3'*3+'2'*2)]

d = set()
for s in ss:
    while '133' in s or '221' in s or '23' in s:
        s = s.replace('133', '1', 1)
        s = s.replace('221', '31', 1)
        s = s.replace('23', '21', 1)
    d.add(s)
print(len(d))
```

Получаем ответ: 640.

