

Найди все целые числа в отрезке [10000000,19999999], все делители которых образуют между собой геометрическую прогрессию, содержащую не менее 5 членов.

В ответе напиши сами числа в порядке возрастания и знаменатель геометрической прогрессии для делителей (во сколько раз следующий член больше предыдущего)

12117361 59

13845841 61

14348907 3

16777216 2

19487171 11

Медленное решение

```
#Стандартный алгоритм для решения 25х задач. С таким диапазоном работает долго, но если есть лишние полчаса - решит
def proverka(x):
    for i in range(len(x)-2):
        if x[i+1] / x[i] != x[i+2] / x[i+1]:
            return False #Если знаменатель прогрессии для членов не совпадает с знаменателем следующих членов - вернет ложь.
    return True #Все совпадает - вернем правду. -2 в начале для того, чтобы не было ошибки о отсутствующих индексах
#-----
def isprime(x):
    sq = int(x**0.5) #Стандартная функция для определения простоты числа. У каждого составного числа есть делитель до
    for i in range(2,sq+1):
        if x%i==0:
            return False #Перебираем числа в диапазоне от 2 до целой части квадратного корня. Делаем проверку на делимость
    return x>1 #Если мы делитель не нашли, то мы проверяем, является ли число единицей. Если является - то число не простое
#-----
for x in range(10000000,20000000): #Перебираем числа в нашем диапазоне.
    sq = int(x**0.5) #Целая часть квадратного корня числа
    d = set() #Множество, куда мы будем забивать делители. Множество потому-что в нем не может быть повторений.
    if isprime(x): #В угоду времени пропускаем простые числа.
        continue
    for i in range(1,sq+1): #Перебираем делители.
        if x%i == 0: #Если мы нашли таковые, то добавляем 2 числа. Само i и x // i в пару.
            d.add(i)
            d.add(x//i)
    d = sorted(d) #Поворачиваем наше множество в отсортированный в порядке возрастания список.
    if proverka(d) and len(d) >= 5: #Если этот список является геом. прогрессией и его длина больше 5, то это наше число.
        print(x,d[1])
```

Быстрое решение

```
#Если ученик попадет на смысленый, он поймет, что от него требуются степени простых чисел.
#Если числа не будут простыми, то в нашу прогрессию будут добавлены лишние числа -
#делители самих членов прогрессии, которые нам не нужны
def isprime(x):
    sq = int(x**0.5)
    for i in range(2,sq+1): #Стандартная функция определения простоты, объяснял в прошлом способе решения
        if x%i==0:
            return False
    return x>1
#-----
a = [x for x in range(1,1000) if isprime(x)] #Фильтруем диапазон, заданный на глазок. Это - наши простые основания.
for osnovanie in a: #Перебираем числа из списка оснований
    for stepen in range(4,30): #В данном случае, stepen+1 - длина нашей прогрессии.
        # +1 потому-что в прогрессии присутствует единица - нулевая степень
        if osnovanie**stepen in range(10000000,20000000): #Основание**степень - наше число. Проверяем, лежит ли оно в диапазоне
            print(osnovanie**stepen,osnovanie) #Если лежит, то оно наше. Выводим число и основание. Основание - знаменатель
```