

strim_25_2

Как искать делители числа?

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=0h1m0s

Для любого целого положительного числа n , делителями называются такие целые положительные числа, на которые n делится нацело, то есть без остатка. Например, для числа $n = 6$, делителями будут числа $\{1, 2, 3\}$. Количество делителей может варьироваться в зависимости от конкретного числа: у одних чисел их больше, у других — меньше. Рассмотрим различные методы нахождения этих делителей.

Метод 1: Проверка всех чисел

Самый простой способ найти все делители числа заключается в проверке всех целых положительных чисел от 1 до n . Рассмотрим этот метод на примере числа $n = 1\,000\,000$.

Сначала создаётся пустой список для хранения найденных делителей. Затем последовательно проверяется каждое целое число от 1 до $1\,000\,000$ на предмет того, делится ли n на это число без остатка. Если результат деления равен нулю, то текущее число добавляется в список делителей.

Алгоритм выглядит следующим образом:

```
x=1_000_000
a=[]
for i in range(1,x+1):
    if x%i==0:
        a.append(i)
print(a)
```

Когда алгоритм завершит свою работу, в списке a окажутся все делители числа $1\,000\,000$.

Однако у этого метода есть существенный недостаток: необходимо проверить очень много чисел, хотя на практике количество делителей обычно гораздо меньше.

Метод 2: Использование парных делителей

Более эффективный подход основан на использовании свойств парных делителей. Если число d является делителем числа n , то n/d также будет делителем. Это позволяет ограничить проверку только числами до корня из n , вместо проверки всех чисел вплоть до n .

Рассмотрим применение этого метода на примере числа $n = 200$:

1. Начинаем с числа 1, которое всегда является делителем. Тогда $200/1 = 200$ и 200 также является делителем.
2. Число 2 также является делителем, и $200/2 = 100$, следовательно, 100 тоже делитель.
3. Число 3 не является делителем 200, поэтому его пропускаем.
4. Число 4 является делителем, и $200/4 = 50$, значит, 50 тоже делитель.

Аналогичным образом продолжаем поиск, находя следующие пары делителей.

Достаточно перебора и сел до корня!

$X = 200$	$X = 225$
1 ; 200	1 ; 225
2 ; 100	3 ; 75
4 ; 50	5 ; 45
5 ; 40	9 ; 25
8 ; 25	15 ; 15
10 ; 20	

Дойдя до числа 20, понимаем, что дальнейшее продолжение поиска становится нецелесообразным, поскольку все оставшиеся делители уже были найдены в виде второй части соответствующих пар. Этот метод значительно уменьшает количество необходимых проверок по сравнению с первым методом, тем самым увеличивая общую производительность.

Таким образом, чтобы найти все делители числа n , достаточно проверить числа от 1 до корня из n . При обнаружении нового делителя d , одновременно определяется второй делитель n/d . Этот подход заметно сокращает количество проверок и ускоряет процесс поиска делителей.

Теперь реализуем этот метод в виде кода. Напишем функцию, которая будет находить все делители числа. Вместо списка используем множество (set), чтобы исключить возможность добавления дубликатов при работе с парными делителями.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)
print(div(100_000_000))
```

Функция `div(x)` находит все делители числа x и возвращает их в отсортированном порядке.

Задача № 1 (2562)

(Демовариант 2021 г.). Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [174457; 174505], числа, имеющие ровно два различных натуральных делителя, не

считая единицы и самого числа. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=0h22m45s

Решение

Для начала напишем функцию `div`, которая будет находить все натуральные делители числа x . Чтобы исключить единицу и само число, будем начинать поиск делителей с числа 2. Также заметим, что при поиске делителей числа x нам достаточно проверять делимость до корня из x , поскольку если число делится на i , оно также делится на x/i .

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)
```

Теперь, используя эту функцию, мы можем перебрать все числа в заданном диапазоне и для каждого числа проверить, имеет ли оно ровно два делителя. Если условие выполняется, выводим эти два делителя в порядке возрастания.

```
for x in range(174457, 174506):
    d = div(x)
    if len(d)==2:
        print(d[0],d[1])
```

Вся программа:

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)
for x in range(174457, 174506):
    d = div(x)
    if len(d)==2:
        print(d[0],d[1])
```

Результат работы программы:

```
3 58153
7 24923
59 2957
```

13 13421
149 1171
5 34897
211 827
2 87251

Ответ:

3	58153
7	24923
59	2957
13	13421
149	1171
5	34897
211	827
2	87251

Задача №2 (2570)

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [135743; 135789], числа, имеющие ровно 6 различных делителей. В ответе для каждого найденного числа запишите два его наибольших делителя в порядке возрастания.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=0h30m35s

Решение

Создадим функцию `div`, которая будет находить все делители числа x . Мы начнем перебирать потенциальные делители с единицы, так как она считается одним из делителей любого числа. При нахождении делителя мы добавляем его и соответствующий ему сопряженный делитель x/i в множество `d`. Это позволит избежать дублирования делителей и упростит дальнейшую работу с ними. После завершения цикла мы сортируем полученное множество делителей.

```
def div(x):  
    d = set()  
    for i in range(1, int(x**0.5)+1):  
        if x%i==0:  
            d.add(i)  
            d.add(x//i)  
    return sorted(d)
```

Далее, мы переберем все числа в диапазоне от 135743 до 135789 включительно. Для каждого числа найдем его делители с помощью функции `div` и проверим, соответствует ли количество делителей шести. Если условие выполнено, выведем два наибольших делителя числа. Для вывода двух наибольших делителей мы используем отрицательные индексы `[-2]` и `[-1]`, которые соответствуют предпоследнему и последнему элементам отсортированного списка делителей. Этот метод универсален и работает независимо от общего количества делителей, что делает его предпочтительным в случае, если количество делителей неизвестно заранее.

```
for x in range(135743,135790):
    d = div(x)
    if len(d)==6:
        print(d[-2], d[-1])
```

Итоговый код:

```
def div(x):
    d = set()
    for i in range(1,int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(135743,135790):
    d = div(x)
    if len(d)==6:
        print(d[-2], d[-1])
```

Результат работы программы:

```
45249 135747
67874 135748
67882 135764
27155 135775
```

Ответ:

45249	135747
67874	135748
67882	135764
27155	135775

Болтаем про округление чисел : https://vk.com/video-205546952_456241274?t=0h38m15s

Функция `int()` не только преобразует строку в число, но и округляет вещественные числа, отбрасывая дробную часть. При этом важно помнить, что округление происходит не по математическим правилам, а путем простого удаления дробной части числа.

Задача № 3 (2844)

Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку $[152346; 957812]$ и имеющие ровно три нетривиальных делителя. Для каждого найденного числа запишите в ответе само число и его наибольший нетривиальный делитель. Найденные числа расположите в порядке возрастания.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=0h45m45s

Решение

Начнем с написания функции `div`, которая находит все делители числа `xx`, исключая единицу и само число. Для этого перебор потенциальных делителей начнется с двойки. Если количество делителей равно трем, то выводим само число и его наибольший нетривиальный делитель.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(152346, 957813):
    d = div(x)
    if len(d)==3:
        print(x, d[-1])
```

Однако, такое решение требует значительного времени для выполнения, так как приходится перебирать большое количество чисел. Попробуем оптимизировать этот процесс.

Обратим внимание, что числа с нечетным количеством делителей встречаются редко. Обычно количество делителей четное, так как они образуют пары. Исключение составляют лишь случаи, когда число является квадратом некоторого другого числа. В этом случае у числа появляется дополнительный делитель, который совпадает со своим сопряженным, и общее количество делителей становится нечетным.

Таким образом, вместо того чтобы перебирать все числа в заданном диапазоне, можно ограничиться перебором только тех чисел, которые являются квадратами других чисел. Это значительно сократит объем вычислений.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
```

```

return sorted(d)

for x in range(152346, 957813):
    d = div(x)
    if len(d)==3:
        print(x, d[-1])

#Оптимизация
for i in range(391, 978+1):
    x = i**2
    d = div(x)
    if len(d)==3:
        print(x, d[-1])

```

Результат работы программы:

```

279841 12167
707281 24389
923521 29791
279841 12167
707281 24389
923521 29791

```

Ответ:

279841	12167
707281	24389
923521	29791
279841	12167
707281	24389
923521	29791

Задача № 4 (2573)

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [190061; 190072], числа, имеющие ровно 4 различных НЕЧЁТНЫХ делителя. В ответе для каждого найденного числа запишите два его наибольших нечётных делителя в порядке убывания.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=0h58m25s

Решение

Первым шагом создадим функцию `div`, которая будет находить все делители числа x . Мы начнем перебор с единицы, так как нам не запрещено учитывать единицу и само число в качестве делителей. Затем проверим каждый возможный делитель от 1 до \sqrt{x} и добавим его в множество `d`, чтобы избежать дублирования.

```
def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)
```

После создания функции для поиска делителей, мы перебираем все числа в заданном диапазоне от 190061 до 190072 включительно. Для каждого числа находим его делители с помощью функции `div` и фильтруем их, оставляя только нечётные делители.

```
for x in range(190061, 190073):
    d = [i for i in div(x) if i%2!=0]
```

Затем проверяем, содержит ли список нечётных делителей ровно четыре элемента. Если это так, выводим два наибольших нечётных делителя данного числа в порядке убывания.

```
if len(d)==4:
    print(d[-1], d[-2])
```

Собрав все шаги вместе, получаем следующий код:

```
def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(190061, 190073):
    d = [i for i in div(x) if i%2!=0]
    if len(d)==4:
        print(d[-1], d[-2])
```

Запустив данный код, мы получим искомые результаты, соответствующие условиям задачи.

Ответ:

190061	6131
11879	1697
190067	2677
23759	1033

Задача №5 (5144)

Пусть $N(k) = 750\,000 + k$, где k – натуральное число. Найдите пять наименьших значений k , при которых $N(k)$ имеет нечётное количество различных чётных делителей. В ответе запишите найденные значения k в порядке возрастания, справа от каждого значения запишите число чётных делителей $N(k)$.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=1h8m30s

Решение

Первым делом создадим функцию `div`, которая будет находить все делители числа x . В данной задаче нет ограничения на использование единицы и самого числа в качестве делителей, поэтому мы будем перебирать все возможные делители от 1 до \sqrt{x} . Поскольку в условии задачи не указан конкретный диапазон значений k , мы будем последовательно увеличивать k , начиная с 1, и проверять каждое значение $N(k)$, пока не найдём пять нужных значений. Для удобства можно начать с определённого диапазона, например, до 10000. Если программа не выдаст нужное количество решений (пять), то следует увеличить границу и повторить попытку. Всех делителей числа $N(k)$ нам нужны только чётные. Используем списковое включение для фильтрации чётных делителей. Затем проверим, является ли количество чётных делителей нечётным. Если это так, выводим текущее значение k и количество чётных делителей.

```
def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for k in range(1, 15000):
    x = 750_000 + k
    d = [i for i in div(x) if i%2==0]
    if len(d)%2!=0:
        print(k, len(d))
```

Для того, чтобы получить требуемое в ответе количество делителей. Нам потребовалось расширить диапазон перебираемых значений до 15000. Из 6 полученных значений в ответ выбираем первые 5 штук

Результат работы программы:

1538 3
3992 9
6450 27
8912 63
11378 3
13848 27

Ответ:

1538	3
3992	9
6450	27
8912	63
11378	3

Задание №6 (4215)

Найдите 5 чисел больших 500000, таких, что среди их делителей есть число, оканчивающееся на 8, при этом этот делитель не равен 8 и самому числу. В качестве ответа приведите 5 наименьших чисел, соответствующих условию. Формат вывода: для каждого из найденных чисел в отдельной строке запишите само число, а затем минимальный делитель, оканчивающийся на 8, не равный 8 и самому числу.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=1h8m30s

Решение

Для решения задачи необходимо найти пять чисел, больших 500 000, у которых среди делителей есть число, оканчивающееся на 8, при этом этот делитель не равен 8 и самому числу.

Первым шагом будет написание функции для нахождения всех делителей числа. Поскольку условие исключает единицу и само число, начнем поиск делителей с двойки. Функция `div` находит все делители числа `x`, используя метод перебора от 2 до квадратного корня из `x`.

Затем переберем числа, начиная с 500001, чтобы найти те, которые удовлетворяют условиям задачи. Для каждого числа проверяем, есть ли среди его делителей такие, которые заканчиваются на 8, но не равны 8. Если такие делители существуют, выводим само число и наименьший подходящий делитель.

```
def div(x):  
    d = set()  
    for i in range(2, int(x**0.5)+1):  
        if x%i==0:  
            d.add(i)  
            d.add(x//i)  
    return sorted(d)
```

```
for x in range(500_001, 500_100):  
    d = [i for i in div(x) if i%10==8 and i!=8]  
    if len(d)>0:  
        print(x,d[0])
```

Результат работы программы:

500002 178

500004 18

500016 48

500018 58

500020 4348

500022 18

500024 28

500028 83338

500032 208

500034 166678

500036 6098

500038 238

500040 18

500042 38

500044 298

500052 28

500056 250028

500058 18

500064 48

500066 71438

500070 158

500072 68

500076 18

500078 8198

500080 28

500084 118

500088 268

500090 100018

500092 148

500094 18

500096 128

В ответ возьмем, по условию, только первые пять пар

Ответ:

500002	178
500004	18
500016	48
500018	58
500020	4348

Задание № 7(7502)

(ЕГЭ-2024) Пусть M – сумма минимального и максимального натуральных делителей целого числа, не считая единицы и самого числа. Если таких делителей у числа нет, то значение M считается равным нулю. Например, для числа 20 имеем $M = 2 + 10 = 12$. Напишите программу, которая перебирает целые числа, меньшие 800 000, в порядке убывания и ищет среди них такие, для которых значение M оканчивается на 2. В ответе запишите в первом столбце таблицы первые пять найденных чисел в порядке возрастания, а во втором столбце – соответствующие им значения M .

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=1h30m10s

Решение

Для начала определимся с алгоритмом нахождения всех делителей числа, за исключением единицы и самого числа. Для этого используем уже известный нам подход:

```
def div(x):  
    d = set()  
    for i in range(2, int(x**0.5)+1):  
        if x%i==0:  
            d.add(i)  
            d.add(x//i)  
    return sorted(d)
```

Теперь, когда функция для поиска делителей готова, приступим к основной части программы. Будем перебирать числа от 799 999 до 800 000 в порядке убывания и вычислять значение M для каждого числа. Если M заканчивается на 2, выведем соответствующее число и значение M .

```

for x in range(799_000,800_000):
    d = div(x)
    if len(d)>0:
        M = d[-1] + d[0]
        if M%10==2:
            print(x,M)

```

Итоговый код:

```

def div(x):
    d = set()
    for i in range(2,int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(799_000,800_000):
    d = div(x)
    if len(d)>0:
        M = d[-1] + d[0]
        if M%10==2:
            print(x,M)

```

Результат работы программы:

799000 399502

799017 266342

799020 399512

799037 27582

799040 399522

.....

799711 72712

799717 15142

799720 399862

Запуская этот код, мы получим список чисел и соответствующих значений М, удовлетворяющих условиям задачи. Однако, согласно заданию, необходимо выбрать первые пять найденных чисел в порядке возрастания. Поэтому в ответ пойдут первые пять элементов.

Ответ:

799000	399502
799017	266342
799020	399512
799037	27582
799040	399522

Задание №8 (4520)

Обозначим через $P(N)$ – произведение 5 наименьших различных нетривиальных делителей натурального числа N (не считая единицы и самого числа). Если у числа N меньше 5 таких делителей, то $P(N)$ считается равным нулю. Найдите 5 наименьших натуральных чисел, превышающих 200 000 000, для которых $P(N)$ оканчивается на 1 и не превышает N . В ответе для каждого найденного числа запишите сначала значение $P(N)$, а затем – наибольший делитель, вошедший в произведение $P(N)$.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=1h37m0s

Решение

Первым шагом является создание функции для нахождения всех делителей числа, исключая единицу и само число. Для этого используется метод перебора возможных делителей от 2 до \sqrt{N} включительно. После того как функция для поиска делителей создана, перейдём к основной части программы. Необходимо перебрать числа начиная с 200 000 001 и выше, находить их делители и вычислять $P(N)$. Если количество делителей достаточно велико (пять и более), вычисляется произведение первых пяти делителей. Далее проверяется, оканчивается ли $P(N)$ на 1 и не превышает ли оно само число N . Если оба условия выполняются, выводятся $P(N)$ и наибольший делитель, участвующий в произведении.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

# Перебираем числа от 200 000 001 до 201 000 000
for x in range(200_000_001, 201_000_000):
    # Находим делители числа x
    d = div(x)
    # Проверяем наличие минимум пяти делителей
    if len(d) >= 5:
        # Вычисляем произведение первых пяти делителей
        P = d[0] * d[1] * d[2] * d[3] * d[4]
        # Проверяем, оканчивается ли P на 1 и не превышает ли оно x
        if P % 10 == 1 and P <= x:
            # Выводим P и наибольший делитель из произведения
            print(P, d[4])
```

Ответ:

5000211	171
5000211	171
391391	23
122124411	227
1712421	81
194481	49

Задание №9(4746)

Пусть $S(N)$ – сумма трёх наибольших нетривиальных делителей числа N (не считая единицы и самого числа). Если у числа N меньше трёх таких делителей, то $S(N)$ считается равным 0. Найдите 5 наименьших натуральных чисел, превышающих 10 000 000, для которых $S(N)$ – полный квадрат какого-либо числа. В ответе запишите найденные числа в порядке возрастания, справа от каждого числа запишите соответствующее ему значение $S(N)$.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=1h44m0s

Решение

Чтобы определить, является ли число n полным квадратом, воспользуемся следующим условием: если квадратный корень из числа nn равен целому числу, то n — это полный квадрат.

Математически это выражается так:

$$(\sqrt{n})^2 = n$$

$$j^2 = n$$

или эквивалентно:

$$(\sqrt{n})^2 \% 1 = 0$$

Это означает, что остаток от деления n^2 на 1 должен быть равен 0.

Перебираем числа, начиная с 10000001, и для каждого числа находим его делители. Если количество делителей больше или равно трём, рассчитываем сумму трёх наибольших делителей и проверяем, является ли эта сумма полным квадратом. Если да, выводим число и соответствующую ему сумму.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(10_000_001, 10_100_000):
    d = div(x)
    if len(d)>=3:
        s = d[-3]+d[-2]+d[-1]
        if s**0.5%1==0:
            print(x, s)
```

Результат работы программы:

10012335 6007401
10012912 8940100

10019645 2582449
10037404 8300161
10045273 2105401
10046428 8970025
10064363 912025
10064496 10903204
10067975 2455489
10096220 9591409

Ответ:

10012335	6007401
10012912	8940100
10019645	2582449
10037404	8300161
10045273	2105401

Задание №10 (7784 ред.)

Пусть $M(N)$ – сумма двух наибольших различных натуральных делителей натурального числа N , не считая самого числа и единицы. Если у числа N меньше двух таких делителей, то $M(N)$ считается равным 0. Найдите все такие числа N , что $256\,123\,000 \leq N \leq 256\,234\,000$, а десятичная запись числа $M(N)$ заканчивается на 1234. В ответе перечислите все найденные числа N , справа от каждого запишите соответствующее значение M . Данные отсортируйте по убыванию значений M .

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241274?t=1h57m20s

Решение

Для начала определимся с задачей: требуется найти все натуральные числа N в диапазоне от 256 123 000 до 256 234 000 включительно, для которых выполняется следующее условие: сумма двух наибольших различных натуральных делителей числа N , исключая само число и единицу, оканчивается на 1234. Эти числа должны быть выведены вместе со значением суммы, причем данные необходимо отсортировать по убыванию значения этой суммы.

Алгоритм решения задачи следующий:

1. Для каждого числа N из указанного диапазона находим все его делители, используя метод перебора от 2 до \sqrt{N} .
2. Из списка делителей выбираем два наибольших, исключая само число и единицу, и вычисляем их сумму.
3. Проверяем, оканчивается ли эта сумма на 1234. Это делается путём взятия остатка от деления суммы на 10000.
4. Если условие выполнено, добавляем пару (N , сумма) в список результатов.
5. Сортируем результаты по убыванию суммы и выводим результат.

Теперь перейдём к реализации алгоритма на Python:

```
# Функция находит все делители числа x
def div(x):
    # Используем множество для хранения уникальных делителей
    d = set()
    for i in range(2, int(x ** 0.5) + 1):
        if x % i == 0:
```



```

        # Добавляем сам делитель
        d.add(i)
        # Добавляем второй делитель
        d.add(x // i)

    # Возвращаем отсортированный список делителей
    return sorted(d)

# Основной цикл поиска подходящих чисел
# Список для хранения пар (сумма, число)
a = []
for n in range(256_123_000, 256_234_001):
    # Получаем список всех делителей числа n
    d = div(n)
    # Убедимся, что у числа есть хотя бы два делителя, кроме самого числа и единицы
    if len(d) >= 2:
        # Сумма двух наибольших делителей
        M = d[-2] + d[-1]
        # Проверяем, оканчивается ли сумма на 1234
        if M % 10000 == 1234:
            # Добавляем пару (сумма, число) в список
            a.append((M, n))

# Сортируем список по убыванию первого элемента (т.е. суммы)
a.sort(reverse=True)
# Выводим результат
for M, n in a:
    print(n, M)

```

Ответ :

256228312	192171234
256148312	192111234
256153623	108671234
256232121	96551234
256182933	85531234
256202495	66311234
256186145	60071234
256164673	41234

Telegram: @fast_ege