

25 (высокий уровень, время – 20 минут)

Тема: Обработка целых чисел. Проверка делимости

Что проверяется:

Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации.

1.5.2. Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности (?).

1.1.3. Строить информационные модели объектов, систем и процессов в виде алгоритмов.

Что нужно знать:

- в известных задачах этого типа (не олимпиадных) нет ограничения на время выполнения, по крайней мере, оно несущественно для отрезков, заданных для перебора; поэтому можно использовать простой перебор без оптимизации;
- задачи этого типа предлагается решать с электронных таблиц или собственной программы; как правило, написать правильную программу значительно проще
- пусть необходимо перебрать все целые числа на отрезке $[a; b]$ и подсчитать, для скольких из них выполняется некоторое условие; общая структура цикла перебора записывается так (Python):

```
count = 0
for n in range(a, b+1):
    if УСЛОВИЕ ВЫПОЛНЕНО:
        count += 1
print( count )
```

Pascal:

```
count := 0;
for n:=a to b do
    if УСЛОВИЕ ВЫПОЛНЕНО then
        count := count + 1;
writeln(count);
```

C++:

```
int count = 0;
for(int n = a; n <= b; n++)
    if( УСЛОВИЕ ВЫПОЛНЕНО )
        count += 1;
std::cout << count;
```

- проверку условия удобно оформить в виде функции, возвращающей логическое значение (True/False), но можно этого и не делать
- проверить делимость числа n на число d можно с помощью операции взятия остатка от деления n на x : если остаток равен 0, число n делится на x нацело
- проверка делимости на языке Python выглядит так:

```
if n % d == 0:
    print("Делится")
else: print("Не делится")
```

- тоже самое на Pascal

```
if n mod d = 0 then
    writeln('Делится')
else writeln('Не делится')
```

- то же самое на C++

```

if( n % d == 0 )
    std::cout << "Делится";
else std::cout << "Не делится";

```

Количество делителей

- для определения числа делителей натурального числа n можно использовать цикл, в котором перебираются все возможные делители d от 1 до n , при обнаружении делителя увеличивается счётчик делителей:

```

count = 0
for d in range(1, n+1):
    if n % d == 0:
        count += 1
print( count ) # вывести количество делителей

```

- тоже самое на Pascal

```

count := 0;
for d:=1 to n do
    if n mod d = 0 then
        count := count + 1;
writeln( count );

```

- то же самое на C++

```

int count = 0;
for(int d = 1; d <= n; d++)
    if( n % d == 0 ) count ++;
std::cout << count; // вывести количество делителей

```

- если требуется определить не только количество делителей, но и сами делители, нужно сохранять их в массиве
- в языке Python удобно использовать динамический массив: сначала он пуст, а при обнаружении очередного делителя этот делитель добавляется в массив:

```

divs = []
for d in range(1,n+1): # перебор всех возможных делителей
    if n % d == 0:      # если нашли делитель d
        divs.append(d) # то добавили его в массив

```

- в языках Pascal и C++ проще обойтись без динамического массива; здесь есть два варианта:
 - 1) выделить массив достаточного размера для хранения всех делителей; например, количество делителей числа n явно не превышает n ;
 - 2) хранить только нужное количество делителей, например, если нас интересуют числа, имеющие 4 делителя, достаточно выделить массив из 4-х элементов, а остальные делители в массив не записывать
- перебор делителей можно оптимизировать, учитывая, что наименьший из пары делителей, таких что $a \cdot b = n$, не превышает квадратного корня из n ; нужно только аккуратно обработать случай, когда число n представляет собой квадрат другого целого числа;
- отметим, что для чисел, которые предлагаются в вариантах заданий, такая оптимизация не обязательна; более того, усложнение программы может привести к дополнительным ошибкам...

Простые числа

- простое число n делится только на 1 и само на себя, причём единица не считается простым числом; таким образом, любое простое число имеет только два делителя
- для определения простоты числа можно считать общее количество его делителей; если их ровно два, то число простое, если не два – не простое:

```

nDel = 0                # количество делителей числа
for d in range(1, n+1): # все возможные делители
    if n % d == 0:
        nDel += 1        # нашли ещё делитель
if nDel == 2:
    print( "Число простое" )
else:
    print( "Число составное" )

```

- работу программы можно ускорить: если уже найдено больше двух делителей, то число не простое и можно досрочно закончить работу цикла с помощью оператора **break**:

```

nDel = 0                # количество делителей числа
for d in range(1, n+1): # все возможные делители
    if n % d == 0:
        nDel += 1        # нашли ещё делитель
        if nDel > 2:      # уже не простое число
            break          # досрочный выход из цикла
if nDel == 2:
    print( "Число простое" )
else:
    print( "Число составное" )

```

- другой вариант – считать количество делителей числа на отрезке $[2; n-1]$; как только хотя бы один такой делитель будет найден, можно завершить цикл, потому что число явно не простое:

```

nDel = 0                # количество делителей на отрезке [2; n-1]
for d in range(2, n):
    if n % d == 0:
        nDel += 1        # нашли делитель
        break            # досрочный выход из цикла
if nDel == 0:
    print( "Число простое" )
else:
    print( "Число составное" )

```

- можно сделать то же самое с помощью логической переменной:

```

prime = True            # сначала считаем число простым
for d in range(2, n):
    if n % d == 0:
        prime = False    # уже не простое
        break            # досрочный выход из цикла
if prime:
    print( "Число простое" )
else:
    print( "Число составное" )

```

- тоже самое на Pascal

```

prime := True;          { сначала считаем число простым }
for d:=2 to n-1 do
    if n mod d = 0 then begin
        prime := False;  { уже не простое }
        break            { досрочный выход из цикла }
    end;
if prime then
    writeln( 'Число простое' )

```

```
else
    writeln( 'Число составное' );
```

- то же самое на C++

```
bool prime = true;           // сначала считаем число простым
for( int d = 2; d <= n-1; d++ )
    if( n % d == 0 ) {
        prime = false; // уже не простое
        break;         // досрочный выход из цикла
    }
if( count == 2 )
    std::cout << "Число простое";
else
    std::cout << "Число составное";
```

- в этом задании обычно предлагаются большие числа, поэтому проверка делимости на все числа от 2 до $n-1$ выполняется очень долго, и на устаревших компьютерах время работы приведённого выше алгоритма может быть слишком велико
- программу можно оптимизировать, если вспомнить, что наименьший из пары делителей, таких что $a \cdot b = n$, не превышает квадратного корня из n ; поэтому можно закончить перебор значением \sqrt{n} , округлив его до ближайшего целого числа; если на отрезке $[2; \sqrt{n}]$ не найден ни один делитель, их нет и на отрезке $[\sqrt{n} + 1, n - 1]$
- следовательно, можно существенно ускорить перебор, изменив конечное значение переменной цикла:

```
for d in range(2, round(sqrt(n))+1):
```

- на языке Pascal:

```
for d:=2 to round(sqrt(n)) do
```

- на языке C++:

```
for( int d = 2; d <= round(sqrt(n)); d++ )
```

Особенности языка Python

- (В. Ялдыгин) при записи больших чисел в Python можно использовать знаки подчеркивания; например, `7_777_777` обозначает то же самое, что и `7777777`.

Пример задания:

Р-03. Найдите все натуральные числа, принадлежащие отрезку $[77\,777\,777; 88\,888\,888]$, у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа, справа от каждого числа запишите его наименьший нечётный делитель, не равный 1.

Решение (программа перебирает числа из отрезка, разбор – В.Н. Шубинкин)

- 1) Отметим, что простой перебор (решение «в лоб») для такой задачи будет работать порядка 20 минут, что неприемлемо в условиях экзамена.
- 2) Чтобы написать эффективный алгоритм, обратимся к математике. Известно, что любое число единственным образом (с точностью до порядка сомножителей) представимо в виде произведения простых чисел: $n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$. Здесь p_i ($i=1, \dots, m$) – различные простые делители, а k_i ($i=1, \dots, m$) – их кратности (натуральные числа).
- 3) Все делители числа (кроме 1) можно получить, взяв произведения всевозможных комбинаций простых множителей. Например, $18 = 2 \cdot 3^2$, поэтому делители числа 18 – это 1 и 2, 3, $2 \cdot 3 = 6$, $3 \cdot 3 = 9$, $2 \cdot 3^2 = 18$.

- 4) Рассмотрим случай, когда в разложение числа на простые множители входит ровно два простых нечётных числа каждое в первой степени: $n = 2^k p_1 p_2$. Тогда число n делится на 1, p_1 , p_2 и произведение $p_1 p_2$, т.е. имеет 4 нечётных делителя. Такой случай нам не подходит.
- 5) Попробуем взять одно из простых чисел во второй степени: $n = 2^k p_1^2 p_2$. Тогда нечётными делителями числа n будут: 1, p_1 , p_2 , p_1^2 , $p_1 p_2$, $p_1^2 p_2$. Это уже 6 делителей. Очевидно, что при увеличении количества нечётных простых делителей мы также получим больше 5 нечётных делителей исходного числа.
- 6) Сделаем ключевой для решения задачи вывод: **если число имеет ровно 5 нечётных делителей, в его разложение на простые множители может входить только 1 нечётное простое число**. Тогда этими делителями будут 1, p , p^2 , p^3 , p^4 , а само число имеет вид $n = 2^k p^4$, где k – натуральное число или ноль, p – нечётное простое число.
- 7) Задача свелась к тому, чтобы перебрать числа из отрезка и, убрав из их разложения на простые множители 2^k , определить является ли то что осталось четвёртой степенью простого числа. Наименьшим простым нечётным делителем, отличным от единицы, будет это простое число. Для определения простоты числа воспользуемся вариантом функции `isPrime()` без вещественных чисел (см. идеи в P-02 и P-01).
- 8) Программа на Python:

```
# функция для определения простоты числа
def isPrime( x ):
    if x <= 1: return False
    d = 2
    while d*d <= x:
        if x % d == 0:
            return False
        d += 1
    return True

start, end = 77777777, 88888888
from math import sqrt
# перебираем все числа из отрезка
for n in range(start, end+1):
    x = n
    # убираем из разложения числа x на простые множители все двойки
    while x % 2 == 0: x //= 2
    # находим корень четвёртой степени из того, что осталось
    qX = round(sqrt(sqrt(x)))
    # проверяем, является ли x четвёртой степенью простого числа
    if isPrime(qX) and qX**4 == x:
        print( n, qX )
```

- 9) Решение на PascalABC.NET с использованием модуля school:

```
uses school;
begin
    var startN := 77777777;
    var endN := 88888888;
    // перебираем числа из отрезка
    for var x0 := startN to endN do begin
        var x := x0;
```

```
// убираем из разложения числа x на простые множители все
// двойки
while x.IsEven do x := x div 2;
// находим корень четвёртой степени из того, что осталось
var qX := round(sqrt(sqrt(x)));
// проверяем, является ли x четвёртой степенью простого числа
if qX.IsPrime and (qX*qX*qX*qX = x) then
    Println( x0, qx );
end;
end.
```

10) Ответ:

```
77900162 79
78074896 47
78675968 7
80604484 67
81920000 5
84934656 3
85525504 17
88529281 97
```

Решение (программа перебирает только простые числа, В.Н. Шубинкин)

- 1) Основная идея решения та же, но теперь будем перебирать не числа из отрезка, а простые числа. Если при возведении нечётного простого числа в четвёртую степень и умножении его на какую-либо степень двойки (в т.ч. нулевую), получится число, входящее в отрезок из условия, то оно пойдёт в ответ.
- 2) Достаточно перебрать все простые числа, не превосходящие корень четвёртой степени из правого конца отрезка

```
start, end = 77777777, 88888888
primes = [2]
for i in range(3, int(end**0.25) + 1, 2):
    flag = True
    for d in range(2, int(i**0.5) + 1):
        if i % d == 0:
            flag = False
            break
    if flag:
        primes.append(i)
```

- 3) В Python существует конструкция **for... else...**. Действия по ветке **else** выполняются в том случае, если цикл завершился без **break**. Зная это, код можно слегка сократить:

```
primes = [2]
for i in range(3, int(end**0.25) + 1, 2):
    for d in range(2, int(i**0.5) + 1):
        if i % d == 0:
            break
    else:
        primes.append(i)
```

- 4) Теперь будем возводить каждое нечётное простое число в четвёртую степень и умножать на два, пока не попадём в нужный отрезок либо не перескочим через него:

```
ans = []
# рассматриваем все простые числа, кроме двойки
```

```

for el in primes[1:]:
    num = el**4
    while num <= end:
        if num >= start:
            ans.append((num, el))
        num *= 2

```

- 5) При таком подходе к решению, ответ необходимо отсортировать, так как мы будем получать значения в порядке возрастания простых чисел, а не чисел из отрезка

```

print(*sorted(ans), sep='\n')

```

- 6) Полное решение на Python:

```

start, end = 77777777, 88888888
primes = [2]
for i in range(3, int(end**0.25) + 1, 2):
    for d in range(2, int(i**0.5) + 1):
        if i % d == 0:
            break
    else:
        primes.append(i)

ans = []
for el in primes[1:]:
    num = el**4
    while num <= end:
        if num >= start:
            ans.append([num, el])
        num *= 2
print(*sorted(ans), sep='\n')

```

- 7) Полное решение на PascalABC.NET

```

begin
    var startN := 77777777; var endN := 88888888;
    // генерируем список простых чисел
    var primes := new List<integer>;
    primes.Add(2);
    foreach var i in range(3, round(endN**0.25) + 1, 2) do
        begin
            var flag := true;
            for var d := 2 to round(i ** 0.5) do
                if i mod d = 0 then begin
                    flag := false;
                    break
                end;
            if flag then primes.add(i)
        end;

    // создаём список кортежей пар чисел, которые пойдут в ответ
    var ans := new List<System.Tuple<real,integer>>;
    // перебираем простые числа, кроме двойки
    foreach var el in primes[1:] do
        begin
            // возводим наши простые числа в четвёртую степень...
            var num := el**4;

```

```
// ...и умножаем на 2, пока они не перескочат через наш отрезок
while num <= endN do
begin
// если при этом число попало в отрезок, помещаем его в ответ
if num >= startN then ans.Add((num, el));
num *= 2
end;
end;
// сортируем полученные значения
ans.Sort();
// выводим на экран ответ
foreach var par in ans do
writeln(par)
end.
```

8) Ответ:

```
(77900162, 79)
(78074896, 47)
(78675968, 7)
(80604484, 67)
(81920000, 5)
(84934656, 3)
(85525504, 17)
(88529281, 97)
```

Ещё пример задания:

Р-02 (демо-2021). 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.

Решение (перебор с помощью программы):

- 11) если число имеет ровно два делителя, отличных от единицы и самого числа, то произведение этих делителей и есть само число; таким образом, строки в таблице должны быть записаны в порядке возрастания чисел, которые они образуют;
- 12) чтобы сами делители в одной строке были записаны в порядке возрастания, нужно выполнять перебор от меньшего числа на отрезке к большему;
- 13) эффективно использовать ускоренный перебор делителей, то есть для числа N перебирать только числа от 2 до $q = \sqrt{N}$ (не включая точный квадратный корень, если он существует); все делители – парные, то есть если a – делитель N , то $b = N/a$ – тоже делитель N
- 14) программа была написана при разборе задачи Р-00, она подходит для любого заданного количества делителей; так как здесь нам нужно выводить все делители, кроме единицы и самого числа, цикле перебора делителей начинаем с 2 и включаем $q = \sqrt{N}$; если очередной делитель d – это точный квадратный корень, добавляем в список делителей только один делитель, если нет – то добавляем пару делителей $(d, x // d)$:

```
from math import sqrt
divCount = 2 # нужное количество делителей
for n in range(174457, 174505+1):
```



```

divs = []
q = int(sqrt(n))
for d in range(2, q+1):
    if n % d == 0:
        if d == n//d:
            divs = divs + [d]
        else:
            divs = divs + [d, n//d]
        if len(divs) > divCount: break
if len(divs) == divCount:
    print( *divs )

```

15) поскольку делителей всего 2, сортировать их не нужно – первым всегда будет меньший из делителей

16) Ответ:

3 58153

7 24923

59 2957

13 13421

149 1171

5 34897

211 827

2 87251

17) программа на Паскале:

```

const divCount = 2;
var n, count, d, i, j, q: longint;
    divs: array[1..divCount] of longint;
begin
    for n:=174457 to 174505 do begin
        count := 0;
        q := floor(sqrt(n));
        for d:=2 to q do
            if n mod d = 0 then begin
                count := count + 2;
                if count <= divCount then begin
                    divs[count-1] := d;
                    if d <> n div d then
                        divs[count] := n div d;
                end
            end
            else break
        end;
        if count = divCount then begin
            for i:=1 to divCount do
                write(divs[i], ' ');
            writeln
        end
    end
end.

```

18) программа на C++:

```

#include <iostream>
#include <cmath>

```

```

int main()
{
    const int divCount = 2;
    int divs[divCount] = {};
    for( int n = 174457; n <= 174505; n++ ) {
        int count = 0;
        int q = int(sqrt(n));
        for( int d = 2; d <= q; d++ )
            if( n % d == 0 ) {
                count += 2;
                if( count <= divCount ) {
                    divs[count-2] = d;
                    if( d != n / d )
                        divs[count-1] = n / d;
                }
                else break;
            }
        if( count == divCount ) {
            for( int i = 0; i < divCount; i++ )
                std::cout << divs[i] << ' ';
            std::cout << std::endl;
        }
    }
}

```

Решение (перебор с помощью программы без использования sqrt):

- 1) при использовании функции для вычисления мы получаем вещественное (не целое) число; вещественные числа (почти всегда) хранятся в памяти компьютера неточно (из-за ограниченного числа разрядов, выделенных на дробную часть), поэтому
 - а) возникает вопрос, куда округлять полученный корень: к меньшему или к большему целому числу?
 - б) нужно аккуратно учесть случай, когда число является полным квадратом другого числа
- 2) для того чтобы вообще избавиться от работы с дробными числами, удобно заменить условие **d <= sqrt(n)** на равносильное условие, использующее только целые значения: **d*d <= n**; при этом, правда, придётся заменить цикл for на while и вручную увеличивать переменную d в конце каждой итерации цикла
- 3) получается такая программа:

```

divCount = 2 # нужное количество делителей
for n in range(174457, 174505+1):
    divs = []
    d = 2
    while d*d <= n:
        if n % d == 0:
            divs.append( d )
            if n//d > d:
                divs.append( n//d )
            if len(divs) > divCount: break
        d += 1
    if len(divs) == divCount:
        print( *divs )

```

- 4) аналогичная программа на Паскале:

```

const divCount = 2;
var n, count, d, i: longint;
    divs: array[1..divCount] of longint;
begin
  for n:=174457 to 174505 do begin
    count := 0;
    d := 2;
    while d*d <= n do begin
      if n mod d = 0 then begin
        count := count + 2;
        if count <= divCount then begin
          divs[count-1] := d;
          if d <> n div d then
            divs[count] := n div d;
          end
        else break
      end;
      d := d + 1;
    end;
    if count = divCount then begin
      for i:=1 to divCount do
        write(divs[i], ' ');
      writeln
    end
  end
end.

```

- 5) программа на C++:

```

#include <iostream>
#include <cmath>
int main()
{
  const int divCount = 2;
  int divs[divCount] = {};
  for( int n = 174457; n <= 174505; n++ ) {
    int count = 0;
    int d = 2;
    while( d*d <= n ) {
      if( n % d == 0 ) {
        count += 2;
        if( count <= divCount ) {
          divs[count-2] = d;
          if( d != n / d )
            divs[count-1] = n / d;
        }
        else break;
      }
      d += 1;
    }
    if( count == divCount ) {
      for( int i = 0; i < divCount; i++ )

```

```

        std::cout << divs[i] << ' ';
        std::cout << std::endl;
    }
}
}

```

Решение (программа без массива, И.В. Степанов):

- 1) учитывая, что в этой задаче нас интересуют только два делителя, можно вместо массива использовать две дополнительных переменные:

```

for i in range (174457, 174505+1):
    k = 0;
    for j in range (2, i):
        if i % j == 0:
            k = k + 1;
            if k == 1: d1 = j
            if k == 2: d2 = j
    if k == 2:
        print( d1, d2 )

```

- 2) вариант программы на Паскале:

```

var i, j, k, d1, d2: longint;
begin
    for i:=174457 to 174505 do begin
        k := 0;
        for j:= 2 to i-1 do
            if i mod j = 0 then begin
                k:= k + 1;
                if k = 1 then d1:= j;
                if k = 2 then d2:= j;
            end;
        if k = 2 then
            writeln( d1, ' ', d2 );
        end;
    end.

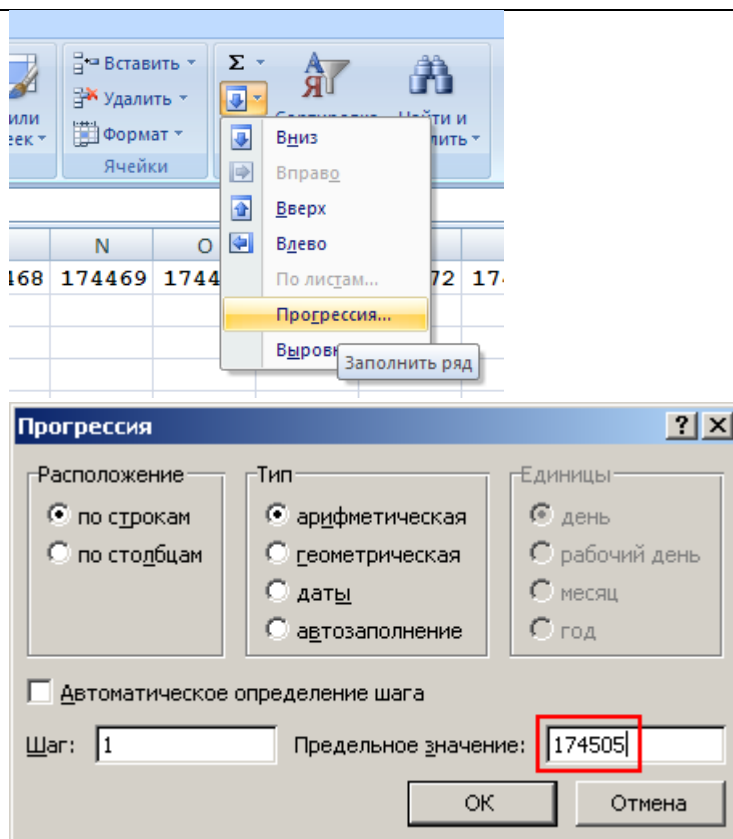
```

Решение (электронные таблицы, И.В. Степанов):

- 1) перебор можно организовать и с помощью электронных таблиц, используя функцию ОСТАТ (MOD) ; для этого в первый столбец занесём все делители от 2 до квадратного корня из наибольшего числа ($\sqrt{174505} \approx 417, \dots$), а в первую строку – все натуральные числа заданного отрезка:

	A	B	C	D	E
1		174457	174458	174459	174460
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				

- 2) в Excel для этого можно использовать команду *Заполнить - Прогрессия*:



- 3) середину таблицы, начиная с B2, заполняем остатками от деления чисел из первой строки на делители из первого столбца;

	A	B	C	D	E
1		174457	174458	174459	174460
2	2	1	0	1	0
3	3	1	2	0	1
4	4	1	2	3	0

В OpenOffice Calc вместо **ОСТАТ** нужно использовать функцию **MOD**

- 4) ниже 417-й строки считаем для каждого числа количество делителей; нас интересуют числа, у которых один делитель на отрезке [2; 417]; используем функцию **СЧЁТЕСЛИ**, с помощью которой считаем нули в каждом столбце (ноль говорит о том, что число из первой строки разделилось нацело на делитель в первом столбце)

	A	B	C	D	E
415	415	157	158	159	160
416	416	153	154	155	156
417	417	151	152	153	154
418		0	3	1	23

В OpenOffice Calc вместо **СЧЁТЕСЛИ** нужно использовать функцию **COUNTIF**

- 5) для тех чисел, у которых всего один делитель, меньший или равный 417, находим его с помощью функции **ПОИСКПОЗ**; она находит в столбце 0 и определяет его позицию (третий аргумент функции **ПОИСКПОЗ** означает точное совпадение):

B419		f_x	=ЕСЛИ(B418=1;ПОИСКПОЗ(0;B1:B417;0);"")				
	A	B	C	D	E	F	G
415	415	157	158	159	160	161	
416	416	153	154	155	156	157	
417	417	151	152	153	154	155	
418		0	3	1	23	1	
419				3		7	

В OpenOffice Calc нужно использовать формулу

=IF(B418=1;MATCH(0;B1:B417;0);"")

- 6) теперь вычисляем второй делитель: делим число в первой строке на первый делитель, всё это только для подходящих чисел:

B420		f_x	=ЕСЛИ(B418=1;B1/B419;"")				
	A	B	C	D	E	F	G
415	415	157	158	159	160	161	
416	416	153	154	155	156	157	
417	417	151	152	153	154	155	
418		0	3	1	23	1	
419				3		7	
420				58153		24923	

В OpenOffice Calc нужно использовать формулу

=IF(B418=1;B1/B419;"")

- 7) теперь остаётся выписать найденные пары делителей
8) Ответ:

3 58153

7 24923

59 2957

13 13421

149 1171

5 34897

211 827

2 87251

Ещё пример задания:

Р-01. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [3532000; 3532160], простые числа. Выведите все найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.

- 1) поскольку нужно вывести не только сами числа, но и их порядковые номера, нужно использовать счётчик:

```
count = 0
for n in range(3532000, 3532160+1):
    if число n простое:
        count += 1
        print( count, n )
```

Решение (простой перебор, может работать очень долго):

- 1) для определения простоты числа ищем полное количество делителей, если оно равно 2, то число простое:

```
count = 0
for n in range(3532000, 3532160+1):
    nDel = 0
    for d in range(1,n+1):
        if n % d == 0: nDel += 1
        if nDel > 2: break
    if nDel == 2:
        count += 1
    print( count, n )
```

- 2) полная программа а языке Pascal:

```
var n, count, d, nDel: longint;
begin
    count := 0;
    for n:=3532000 to 3532160 do begin
        nDel := 0;
        for d:=1 to n do
            if n mod d = 0 then begin
                nDel := nDel + 1;
                if nDel > 2 then break;
            end;
        if nDel = 2 then begin
            count := count + 1;
            writeln( count, ' ', n )
        end
    end
end.
```

- 3) полная программа а языке C++:

```
#include <iostream>
#include <cmath>
int main()
{
    int count = 0;
    for( int n = 3532000; n <= 3532160; n++ ) {
        int nDel = 0;
        for( int d = 1; d <= n; d++ )
            if( n % d == 0 ) {
                nDel += 1;
                if( nDel > 2 ) break;
            }
        if( nDel == 2 ) {
            count++;
            std::cout << count << ' ' << n << std::endl;
        }
    }
}
```

- 4) Ответ:

1 3532007

2 3532019

3 3532021
 4 3532033
 5 3532049
 6 3532091
 7 3532103
 8 3532121
 9 3532147

Решение (ускорение перебора, А.Н. Носкин):

- 1) идея ускорения времени выполнения программы состоит в том, что все простые числа, кроме 2 являются нечетными числами;
- 2) тогда внешний цикл надо начинать не с числа 3532000, а с числа 3532001, при этом шаг цикла составит +2. Окончанием цикла также можно сделать не число 3532160, а 3532159;
- 3) внутренний цикл также должен иметь шаг +2
- 4) приведем полную программу:

```
count = 0
for n in range(3532001, 3532159+1, 2):
    nDel = 0
    for d in range(1, n+1, 2):
        if n % d == 0: nDel += 1
        if nDel > 2: break
    if nDel == 2:
        count += 1
    print( count, n )
```

Решение (ускорение перебора, перебор до \sqrt{n}):

- 1) полная программа на языке Python:

```
from math import sqrt
count = 0
for n in range(3532000, 3532160+1):
    prime = True
    for d in range(2, round(sqrt(n))):
        if n % d == 0:
            prime = False
            break
    if prime:
        count += 1
    print( count, n )
```

- 2) **(Б.С. Михлин)** ещё один вариант, в котором вместо функции sqrt используется возведение в степень 0,5:

```
count = 0
for n in range(3532000, 3532160+1):
    for d in range(2, round(n**0.5)+1):
        if n % d == 0:
            break
    else: # else относится к циклу "for d ...", а не к "if"
        # блок выполняется, если не сработал "break"
        count += 1
    print( count, n )
```


- 3) (Б.С. Михлин) компактное решение, использующее встроенную функцию **all** – она возвращает логическое значение **True**, если все элементы переданного ей списка равны **True**; возвращает **False**, если хотя бы один из них равен **False**:

```
count=0
for n in range(3532000,3532160+1):
    # если у 'n' нет делителей от 2 до корня из n
    # (т.е. все 'd' дают остаток отличный от нуля):
    if all( n%d!=0 for d in range(2,round(n**0.5)+1) ):
        count+=1
        print(count,n)
```

- 4) вариант с функцией **isPrime**, которая возвращает логическое значение **True** (истина) для простых чисел и **False** (ложь) для составных:

```
from math import sqrt
def isPrime(n):
    for d in range(2, round(sqrt(n)+1) ):
        if n % d == 0:
            return False
    return True

count = 0
for n in range(3532000, 3532160+1):
    if isPrime(n):
        count += 1
    print( count, n )
```

- 5) Ответ:

```
1 3532007
2 3532019
3 3532021
4 3532033
5 3532049
6 3532091
7 3532103
8 3532121
9 3532147
```

Решение (программа на языке Pascal):

- 5) обратим внимание на заданный отрезок [3532000; 3532160]; числа в нём превышают 32767 – предел для 16-битных целых чисел типа **integer**; поэтому для того, чтобы программа работала правильно на всех системах, вместо **integer** используем тип **longint**, такие переменные всегда занимают 4 байта (диапазон от –2147483648 до 2147483647)
- 6) для каждого числа **n** из заданного диапазона в цикле ищем делители; количество найденных простых чисел хранится в переменной **count**:

```
var n, count, d: longint;
    prime: boolean;
begin
    count := 0;
    for n:=3532000 to 3532160 do begin
        prime := True;
        for d:=2 to round(sqrt(n)) do
            if n mod d = 0 then begin
```

```

        prime := False;
        break;
    end;
    if prime then begin
        count := count + 1;
        writeln( count, ' ', n )
    end
end
end.

```

- 7) вариант с функцией **isPrime**, которая возвращает логическое значение **True** (истина) для простых чисел и **False** (ложь) для составных:

```

var n, count: longint;
function isPrime( n: integer ): boolean;
var d: longint;
begin
    isPrime := True;
    for d:=2 to round(sqrt(n)) do
        if n mod d = 0 then begin
            isPrime := False;
            break;
        end;
    end;
end;
begin
    count := 0;
    for n:=3532000 to 3532160 do begin
        if isPrime(n) then begin
            count := count + 1;
            writeln( count, ' ', n )
        end
    end
end.

```

- 6) Ответ:

```

1 3532007
2 3532019
3 3532021
4 3532033
5 3532049
6 3532091
7 3532103
8 3532121
9 3532147

```

Решение (программа на языке C++):

- 8) для того, чтобы использовать математические функции. нужно подключить заголовочный файл **cmath**:

```
#include <cmath>
```

- 9) полная программа на языке C++:

```

#include <iostream>
#include <cmath>
int main()

```

```

{
    int count = 0;
    for( int n = 3532000; n <= 3532160; n++ ) {
        bool prime = true;
        for( int d = 2; d <= round(sqrt(n)); d++ )
            if( n % d == 0 ) {
                prime = false;
                break;
            }
        if( prime ) {
            count++;
            std::cout << count << ' ' << n << std::endl;
        }
    }
}

```

- 10) вариант с функцией `isPrime`, которая возвращает логическое значение `true` (истина) для простых чисел и `false` (ложь) для составных:

```

#include <iostream>
#include <cmath>
bool isPrime( int n )
{
    bool prime = true;
    for( int d = 2; d <= round(sqrt(n)); d++ )
        if( n % d == 0 ) {
            prime = false;
            break;
        }
    return prime;
}
int main()
{
    int count = 0;
    for( int n = 3532000; n <= 3532160; n++ )
        if( isPrime(n) ) {
            count++;
            std::cout << count << ' ' << n << std::endl;
        }
}

```

- 11) Ответ:

```

1 3532007
2 3532019
3 3532021
4 3532033
5 3532049
6 3532091
7 3532103
8 3532121
9 3532147

```

Ещё пример задания:

P-00. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [194455; 194500], числа, имеющие ровно 4 различных делителя. Выведите эти четыре делителя для каждого найденного числа в порядке возрастания.

Решение (простой перебор):

- 1) поскольку заданный отрезок [194455; 194500] содержит не так много чисел, можно решать задачу простым перебором, особо не заботясь об эффективности вычислений
- 2) при написании программы на языке Python можно поступить так

```
for для всех чисел n в интервале:
    divs = массив всех делителей n
    if len(divs) == 4:
        вывести массив делителей
```

- 3) полная программа на языке Python:

```
for n in range(194455, 194500+1):
    divs = []
    for d in range(1, n+1):
        if n % d == 0:
            divs.append(d)
    if len(divs) == 4:
        print( *divs )
```

- 4) (Б.С. Михлин) построить массив делителей на языке Python можно и с помощью генератора списка:

```
for n in range(194455, 194500+1):
    divs = [d for d in range(1, n+1) if n % d == 0]
    if len(divs) == 4:
        print( *divs )
```

Аналогично можно построить массив делителей, удовлетворяющих заданному условию, например, всех чётных делителей:

```
for n in range(194455, 194500+1):
    divs = [d for d in range(1, n+1)
            if n % d == 0 and d % 2 == 0]
    if len(divs) == 4:
        print( *divs )
```

К сожалению, этот способ сложно использовать в других языках программирования.

- 5) в качестве оптимизации можно прерывать работу внутреннего цикла, когда найден пятый делитель (число n уже точно не подходит), но это не критично:

```
for n in range(194455, 194500+1):
    divs = []
    for d in range(1, n+1):
        if n % d == 0:
            divs.append(d)
            if len(divs) > 4: break
    if len(divs) == 4:
        print( *divs )
```

- 6) ещё один вариант программы (с функцией, которая возвращает массив делителей):

```
def allDivisors(n):
    divs = []
    for d in range(1, n+1):
        if n % d == 0:
```

```

        divs.append(d)
    return divs

for n in range(194455, 194500+1):
    divs = allDivisors(n)
    if len(divs) == 4:
        print( *divs )

```

7) Ответ:

```

1 5 38891 194455
1 163 1193 194459
1 139 1399 194461
1 2 97231 194462
1 113 1721 194473
1 439 443 194477
1 2 97241 194482
1 43 4523 194489
1 11 17681 194491

```

Решение (ускоренный перебор):

- идея состоит в том, чтобы для определения количества делителей числа N перебирать только числа до $q = \sqrt{N}$; если число q целое, его нужно добавить в список делителей, а все остальные делители – парные, то есть если a – делитель N , то $b = N/a$ – тоже делитель N
- получается такая программа, которая подходит для любого заданного количества делителей:

```

from math import sqrt
divCount = 4 # нужное количество делителей
for n in range(194455, 194500+1):
    divs = []
    q = round(sqrt(n))
    if q*q == n:
        divs = [q]
        q -= 1
    for d in range(1, q+1):
        if n % d == 0:
            divs = divs + [d, n//d]
            if len(divs) > divCount: break
    if len(divs) == divCount:
        print( *sorted(divs) )

```

3) Ответ:

```

1 5 38891 194455
1 163 1193 194459
1 139 1399 194461
1 2 97231 194462
1 113 1721 194473
1 439 443 194477
1 2 97241 194482
1 43 4523 194489
1 11 17681 194491

```

Решение (программа на языке Pascal):

- 1) обратим внимание на заданный отрезок [194455; 194500]; числа в нём превышают 32767 – предел для 16-битных целых чисел типа **integer**; поэтому для того, чтобы программа работала правильно на всех системах, вместо **integer** используем тип **longint**, такие переменные всегда занимают 4 байта (диапазон от –2147483648 до 2147483647)
- 2) поскольку нас интересуют только числа, у которых 4 делителя, можно хранить в памяти только первые 4 найденных делителя, а как только будет найден пятый, заканчивать поиск делителей (число нам точно не подходит); такой подход позволяет отказаться от использования динамических массивов и выделить один массив из 4 элементов:

```
divs: array[1..4] of longint;
```

- 3) для каждого числа **n** из заданного диапазона в цикле ищем делители; количество найденных делителей хранится в переменной **count**:

```
count := 0;
for d:=1 to n do                                { перебор всех возможных делителей }
  if n mod d = 0 then begin { нашли делитель }
    count := count + 1;
    if count <= 4 then                                { сохраняем первые 4 делителя }
      divs[count] := d
    else break                                { нашли пятый => выходим }
  end;
```

- 4) полная программа на языке Pascal:

```
var n, count, d, i: longint;
divs: array[1..4] of longint;
begin
  for n:=194455 to 194500 do begin
    count := 0;
    for d:=1 to n do
      if n mod d = 0 then begin
        count := count + 1;
        if count <= 4 then
          divs[count] := d
        else break
      end;
    if count = 4 then begin
      for i:=1 to 4 do
        write(divs[i], ' ');
      writeln
    end
  end.
```

- 5) вариант с функцией **divsNumber**, которая возвращает количество делителей числа:

```
var n, i: longint;
divs: array[1..4] of longint;
function divsNumber(n: longint): longint;
var count, d: integer; { локальные переменные }
begin
  count := 0;
  for d:=1 to n do
    if n mod d = 0 then begin
      count := count + 1;
      if count <= 4 then
```

```

        divs[count] := d
    else break
end;
divsNumber := count
end;
begin
    for n:=194455 to 194500 do begin
        if divsNumber(n) = 4 then begin
            for i:=1 to 4 do
                write(divs[i], ' ');
            writeln
        end
    end;
end.

```

- 8) ускоренный перебор (до $q = \sqrt{N}$); в отличие от программы на Python, нужно вручную делать сортировку массива, поскольку делители записывались в массив не в порядке возрастания:

```

const divCount = 4;
var n, count, d, i, j, q: longint;
    divs: array[1..divCount] of longint;
begin
    for n:=194455 to 194500 do begin
        count := 0;
        q := round(sqrt(n));
        if q*q = n then begin
            count := count + 1;
            divs[count] := q;
            q := q - 1;
        end;
        for d:=1 to q do
            if n mod d = 0 then begin
                count := count + 2;
                if count <= divCount then begin
                    divs[count-1] := d;
                    divs[count] := n div d;
                end
            else break
        end;
        if count = divCount then begin
            { сортировка массива divs }
            for i:=1 to divCount do
                for j:=i to divCount-1 do
                    if divs[j] > divs[j+1] then
                        swap( divs[j], divs[j+1] );
                for i:=1 to divCount do
                    write(divs[i], ' ');
                writeln
            end
        end
    end.
end.

```

- 9) Ответ:

```

1 5 38891 194455
1 163 1193 194459
1 139 1399 194461
1 2 97231 194462
1 113 1721 194473
1 439 443 194477
1 2 97241 194482
1 43 4523 194489
1 11 17681 194491

```

Решение (программа на языке C++):

- 1) при программировании на языке C++ нужно не забыть, что нумерация элементов массивов начинается с нуля
- 2) полная программа на языке C++:

```

#include <iostream>
int main()
{
    int divs[4] = {};
    for( int n = 194455; n <= 194500; n++ ) {
        int count = 0;
        for( int d = 1; d <= n; d++ )
            if( n % d == 0 ) {
                count ++;
                if( count <= 4 )
                    divs[count-1] = d;
                else break;
            }
        if( count == 4 ) {
            for( int i = 0; i < 4; i++ )
                std::cout << divs[i] << ' ';
            std::cout << std::endl;
        }
    }
}

```

- 3) вариант с функцией `divsNumber`, которая возвращает количество делителей числа и заполняет переданный ей массив первыми 4-мя делителями:

```

#include <iostream>
int divsNumber( int n, int divs[] )
{
    int count = 0;
    for( int d = 1; d <= n; d++ )
        if( n % d == 0 ) {
            count ++;
            if( count <= 4 )
                divs[count-1] = d;
            else break;
        }
    return count;
}
int main()

```



```

{
    int divs[4] = {};
    for( int n = 194455; n <= 194500; n++ ) {
        if( divsNumber(n, divs) == 4 ) {
            for( int i = 0; i < 4; i++ )
                std::cout << divs[i] << ' ';
            std::cout << std::endl;
        }
    }
}

```

- 4) ускоренный перебор (до $q = \sqrt{N}$); в отличие от программы на Python, нужно вручную делать сортировку массива, поскольку делители записывались в массив не в порядке возрастания:

```

#include <iostream>
#include <cmath>
int main()
{
    const int divCount = 4;
    int divs[divCount] = {};
    for( int n = 194455; n <= 194500; n++ ) {
        int count = 0;
        int q = round(sqrt(n));
        if( q*q == n ) {
            divs[count] = q;
            count++;
            q -= 1;
        }
        for( int d = 1; d <= q; d++ )
            if( n % d == 0 ) {
                count += 2;
                if( count <= divCount ) {
                    divs[count-2] = d;
                    divs[count-1] = n / d;
                }
                else break;
            }
        if( count == divCount ) {
            // сортировка массива divs
            for( int i = 0; i < divCount; i++ )
                for( int j = i; j < divCount-1; j++ )
                    if( divs[j] > divs[j+1] ) {
                        int temp = divs[j];
                        divs[j] = divs[j+1];
                        divs[j+1] = temp;
                    }
            for( int i = 0; i < divCount; i++ )
                std::cout << divs[i] << ' ';
            std::cout << std::endl;
        }
    }
}

```

5) Ответ:

```
1 5 38891 194455
1 163 1193 194459
1 139 1399 194461
1 2 97231 194462
1 113 1721 194473
1 439 443 194477
1 2 97241 194482
1 43 4523 194489
1 11 17681 194491
```

- 6) (Муфаззалов Д.Ф., г. Уфа) Ускоренный перебор на языке C++ можно осуществлять без сортировки, если располагать делители в нужном порядке по мере их получения; без извлечения корня и округления, если преобразовать неравенство $d < \sqrt{n}$ по правилам математики; и без проверки на полный квадрат, так как если число является полным квадратом, то оно имеет нечетное количество делителей.

```
#include <iostream>
int main()
{
    const int divCount =4;
    int divs[divCount],i,d;
    for( int n = 194455; n <= 194500; n++ )
    {
        int count = 0;
        for( d = 1; d*d < n; d++ )
            if( n % d == 0 )
            {
                divs[count/2] = d;
                divs[divCount-count/2-1]=n/d;
                count+=2;
                if( count > divCount ) break;
            }
        if (count == divCount)
        {
            for( i = 0; i < divCount; i++ )
                std::cout << divs[i] << ' ';
            std::cout << std::endl;
        }
    }
}
```

- 7) (Муфаззалов Д. Ф., г. Уфа) Сортировки можно избежать и если хранить только половину меньших делителей, а другую половину получать при выводе:

```
#include <iostream>
int main()
{
    const int divCount = 4;
    int divs[divCount/2],i,d;
    for( int n = 194455; n <= 194500; n++ ) {
        int count = 0;
```

```

    for( d = 1; d*d < n; d++ )
        if( n % d == 0 ) {
            divs[count/2] = d;
            count += 2;
            if( count > divCount ) break;
        }
    if (count == divCount) {
        for( i = 0; i < divCount/2; i++ )
            std::cout << divs[i] << ' ';
        for( i--; i>=0; i-- )
            std::cout << n/divs[i] << ' ';
        std::cout << std::endl;
    }
}
}

```

- 8) (Мүфаззалов Д. Ф., г. Уфа) Программа с ускоренным перебором, не зависящая от четности количества делителей

```

#include <iostream>
int main()
{
    const int divCount =4;
    int divs[divCount],i,d;
    for( int n = 194455; n <= 194500; n++ ) {
        int count = 0;
        for( d = 1; d*d < n; d++ )
            if( n % d == 0 ) {
                divs[count/2] = d;
                count+=2;
                if( count > divCount ) break;
            }
        if( d*d == n ) {
            divs[count/2] = d;
            count++;
        }
        if (count == divCount) {
            for( i = 0; i < divCount/2; i++ )
                std::cout << divs[i] << ' ';
            if( divCount % 2 )
                std::cout << divs[divCount/2] << ' ';
            for( i--; i>=0; i-- )
                std::cout << n/divs[i] << ' ';
            std::cout << std::endl;
        }
    }
}

```

- 9) В задаче на поиск чисел с четырьмя делителями массив не нужен вовсе. Для таких чисел достаточно найти минимальный делитель, отличный от единицы, а остальные будут равны единице, самому числу и частному от деления самого числа на найденный делитель. Если число имеет хотя бы 2 делителя больше единицы и меньше корня из этого числа, то оно не имеет ровно 4 делителя.

```
#include <iostream>
using namespace std;
int main()
{
    int div,d,count;
    for( int n = 194455; n <= 194500; n++ ) {
        for( d = 2,count = 0; d*d < n && count < 2; d++ )
            if( n % d == 0 && !count++ ) div = d;
        if (count == 1)
            cout << 1 << ' ' << div << ' '
                << n/div << ' ' << n << endl;
    }
}
```


-
- 31) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [1820348; 2880927], числа, имеющие ровно 5 различных делителей. Выведите эти делители для каждого найденного числа в порядке возрастания.
- 32) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [394441; 394505], числа, имеющие максимальное количество различных делителей. Если таких чисел несколько, то найдите **минимальное** из них. Выведите количество делителей найденного числа и два наибольших делителя в порядке убывания.
- 33) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [286564; 287270], числа, имеющие максимальное количество различных делителей. Если таких чисел несколько, то найдите **максимальное** из них. Выведите количество делителей найденного числа и два наибольших делителя в порядке убывания.
- 34) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [586132; 586430], числа, имеющие максимальное количество различных делителей. Найдите **минимальное** и **максимальное** из таких чисел. Для каждого из них в отдельной строчке выведите количество делителей и два наибольших делителя в порядке убывания.
- 35) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [394480; 394540], числа, имеющие максимальное количество различных делителей. Выведите информацию о таких числах, расположив их в порядке возрастания. Для каждого числа выведите его порядковый номер, количество делителей и два наибольших делителя в порядке убывания.
- 36) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [194441; 196500] числа (в порядке возрастания) с нечётным количеством делителей. Для каждого такого числа выведите его порядковый номер (начиная с единицы), само число, количество его делителей и делитель, квадрат которого равен этому числу.
- 37) **(Б.С. Михлин)** Напишите программу, которая ищет среди нечётных целых чисел, принадлежащих числовому отрезку [248015; 251575] числа (в порядке возрастания) с нечётным количеством делителей. Для каждого такого числа выведите его порядковый номер (начиная с единицы), само число, количество его делителей и делитель, квадрат которого равен этому числу.
- 38) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [268220; 270335] число с максимальной суммой делителей, имеющее не более четырех делителей. Для найденного числа выведите сумму делителей, количество делителей и все делители в порядке убывания.
- 39) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [573213; 575340] число с минимальной суммой делителей, имеющее ровно четыре делителя. Для найденного числа выведите сумму делителей и наибольший нетривиальный делитель (не равный самому числу).
- 40) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [2943444; 2943529], простые числа. Выведите все найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.
- 41) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [4671032; 4671106], простые числа. Выведите все найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.
- 42) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [4202865; 4202923], простые числа. Выведите все найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.
-

-
- 32
<http://kpolyakov.spb.ru>

- 58) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [6080068; 6080176], простые числа. Выведите все найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.
- 59) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [7178551; 7178659], простые числа. Выведите все найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.
- 60) **(А.Н. Носкин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [3532000; 3532160], простые числа. Выведите все найденные простые числа в порядке убывания, слева от каждого числа выведите его номер по порядку.
- 61) **(А.Н. Носкин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [2532000; 2532160], простые числа. Выведите все найденные простые числа в порядке убывания, слева от каждого числа выведите его номер по порядку.
- 62) **(А.Н. Носкин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [1532040; 1532160], простые числа. Выведите все найденные простые числа в порядке убывания, слева от каждого числа выведите его номер по порядку.
- 63) **(А.Н. Носкин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [2532000; 2532160] первые пять простых чисел. Выведите найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.
- 64) **(А.Н. Носкин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [2532000; 2532160], простые числа. Найдите все простые числа, которые заканчиваются на цифру 7. Выведите их в порядке возрастания, слева от каждого числа выведите его номер по порядку.
- 65) **(А.Н. Носкин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [2532000; 2532160], простые числа. Найдите все простые числа, но выведите на экран только каждое третье простое число (то есть числа с порядковыми номерами 1, 4, 7, 10, ...). Вывод осуществите в порядке возрастания, слева от каждого числа выведите его собственный порядковый номер среди всех простых чисел.
- 66) **(Б.С. Михлин)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [194441; 196500] простые числа (т.е. числа у которых только два делителя: 1 и само число), оканчивающиеся на 93. Для каждого простого числа выведите его порядковый номер (начиная с единицы), а затем – само число.
- 67) **(П.Е. Финкель, г. Тимашевск)** Уникальным назовём число, если у него только третья и пятая цифры чётные. Для интервала [33333;55555] найдите количество таких чисел, которые не делятся на 6, 7, 8 и разность максимального и минимального из них. В ответе укажите два числа: сначала количество чисел, а потом разность.
- 68) **(П.Е. Финкель, г. Тимашевск)** Уникальным назовём число, если у него только первые две цифры нечётные. Для интервала [57888;74555] найдите количество таких чисел, которые не делятся на 7, 9, 13, и разность максимального и минимального из них. В ответе укажите два числа: сначала количество чисел, а потом разность.
- 69) **(П.Е. Финкель, г. Тимашевск)** Уникальным назовём число, если у него только последние три цифры нечётные. Для интервала [64444;77563] найдите количество таких чисел, которые не делятся на 9, 13, 17, и разность максимального и минимального из них. В ответе укажите два числа: сначала количество чисел, а потом разность.
- 70) **(Д.Ф. Муфаззалов, г. Уфа)** Совершенным называется число, натуральное число, равное сумме всех своих собственных делителей (то есть всех положительных делителей, отличных от самого числа) (например, число $6=1+2+3$).) Выведите каждое совершенное число из диапазона [2; 10000] и

количество его собственных делителей в порядке возрастания. Вывод каждого совершенного числа начинайте с новой строки. Числа в строке разделяйте пробелом.

- 71) (Д.Ф. Муфаззалов, г. Уфа) Определите количество составных натуральных чисел из диапазона [2; 20000], у которых количество простых собственных делителей больше трех.
- 72) (Д.Ф. Муфаззалов, г. Уфа) Найдите в диапазоне [2; 20000] числа, каждое из которых имеет максимальное количество простых делителей среди всех таких чисел. Выведите минимальное из таких чисел и через пробел количество его простых делителей.
- 73) (Д.Ф. Муфаззалов, г. Уфа) Число называется избыточным, если оно меньше суммы своих собственных делителей (то есть всех положительных делителей, отличных от самого числа). Определите количество избыточных чисел из диапазона [2; 20000].
- 74) (Д.Ф. Муфаззалов, г. Уфа) Число называется недостаточным, если оно больше суммы своих собственных делителей (то есть всех положительных делителей, отличных от самого числа). Определите количество недостаточных чисел из диапазона [2; 30000].
- 75) (Д.Ф. Муфаззалов, г. Уфа) Выведите каждое почти совершенное число из диапазона [1000; 20000] в порядке возрастания по одному в строке. Число называется почти совершенным, если оно больше суммы своих собственных делителей (то есть всех положительных делителей, отличных от самого числа) на единицу.
- 76) (Д.Ф. Муфаззалов, г. Уфа) Два числа называются дружественными если сумма собственных делителей (то есть всех положительных делителей, отличных от самого числа) любого из них равна другому числу. Например, числа 220 и 284 дружественные. Выведите в порядке возрастания числа в диапазоне [2; 30000], имеющие дружественное число, большее чем само это число, и через пробел это дружественное число. Каждое следующее число из указанного диапазона выводите на новой строке.
- 77) (Д.Ф. Муфаззалов, г. Уфа) Определите количество простых чисел в диапазоне [2; 20000].
- 78) (Д.Ф. Муфаззалов, г. Уфа) Определите количество простых чисел в диапазоне [2; 200000].
- 79) (Д.Ф. Муфаззалов, г. Уфа) Определите количество простых чисел в диапазоне [2; 3577000].
- 80) (Д.Ф. Муфаззалов, г. Уфа) Найдите в диапазоне [2; 10000000] числа, каждое из которых имеет максимальное количество простых делителей среди всех чисел этого отрезка. Выведите минимальное из найденных чисел и через пробел количество его простых делителей.
- 81) (Д.Ф. Муфаззалов, г. Уфа) Число называется суперсовершенным, если сумма всех делителей суммы всех его делителей равна произведению самого числа на 2. например, число 16 суперсовершенное. Его делители: 1, 2, 4, 8, 16. Их сумма равна 31. Делители числа 31: $1+31=32$. $32=16*2$. Выведите каждое суперсовершенное число из диапазона [2; 263000] в порядке возрастания по одному в строке.
- 82) (Д.Ф. Муфаззалов, г. Уфа) Число называется полусовершенным, если сумма всех или некоторых его собственных делителей (то есть всех положительных делителей, отличных от самого числа) совпадает с самим этим числом. Выведите все полусовершенные числа из диапазона [300; 350] в порядке возрастания по одному в строке.
- 83) (Д.Ф. Муфаззалов, г. Уфа) Число называется полусовершенным, если сумма всех или некоторых его собственных делителей (то есть всех положительных делителей, отличных от самого числа) совпадает с самим этим числом. Определите количество полусовершенных чисел в диапазоне [2; 2000].
- 84) (С.А. Скопинцева) Среди целых чисел, принадлежащих числовому отрезку [87921; 88187], найдите числа, сумма цифр которых кратна 14, а произведение цифр кратно 18 и не равно 0. Для каждого найденного числа запишите сумму и произведение его цифр в таблицу на экране с новой строки в порядке возрастания произведения цифр.

- 85) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [3661; 33625], найдите числа, имеющие ровно один натуральный делитель, не считая единицы и самого числа. Ответом будет количество найденных чисел.
- 86) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [4986; 32599], числа, имеющие ровно два различных натуральных делителя, не считая единицы и самого числа. Ответом будет сумма найденных чисел.
- 87) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [2945; 18294], найдите числа, не делящиеся на вторую степень какого-либо числа, кроме единицы. Ответом будет сумма цифр найденных чисел.
- 88) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [2031; 14312], найдите числа, которые не содержат цифру 2, если записать их в системе счисления с основанием 11. Ответом будет максимум среди найденных чисел.
- 89) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [2948; 20194], найдите числа, которые являются простыми. Ответом будет максимум среди найденных чисел.
- 90) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [3594; 21891], найдите числа, имеющие ровно два различных натуральных делителя, не считая единицы и самого числа. Ответом будет максимум среди найденных чисел.
- 91) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [4099; 26985], найдите числа, имеющие ровно один натуральный делитель, не считая единицы и самого числа. Ответом будет сумма цифр найденных чисел.
- 92) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [1060; 18813], найдите числа, которые являются простыми. Ответом будет сумма найденных чисел.
- 93) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [1686; 13276], найдите числа, все цифры которых нечетные. Ответом будет сумма цифр найденных чисел.
- 94) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [3159; 31584], найдите числа, которые являются простыми. Ответом будет сумма цифр найденных чисел.
- 95) (К. Амеличев) Среди целых чисел, принадлежащих числовому отрезку [1395; 22717], найдите числа, все цифры которых расположены в порядке неубывания. Ответом будет сумма найденных чисел.
- 96) (Е. Джобс) Среди целых чисел, принадлежащих числовому отрезку [81234; 134689], найдите числа, имеющие ровно три различных натуральных делителя, не считая единицы и самого числа. Для каждого найденного числа запишите в таблицу на экране с новой строки сначала наименьший, а потом наибольший из этих делителей.
- 97) (Е. Джобс) Среди целых чисел, принадлежащих числовому отрезку [135790; 163228], найдите числа, сумма натуральных делителей которых больше 460000. Для каждого найденного числа запишите количество делителей и их сумму. В качестве делителей не рассматривать числа 1 и исследуемое число. Так, например, для числа 8 учитываются только делители 2 и 4.
- 98) (Е. Джобс) Среди целых чисел, принадлежащих числовому отрезку [228224; 531135], найдите числа, среди делителей которых есть хотя бы 4 различных куба натуральных нечетных чисел. Для каждого найденного числа запишите в ответе само число, количество таких делителей и наибольший из них. В качестве делителей не рассматривать число 1 и само исследуемое число. Так, например, для числа 8 учитываются только делители 2 и 4.
- 99) (Е. Джобс) Среди целых чисел, принадлежащих числовому отрезку [333555; 777999], найдите числа, среди делителей которых есть ровно 35 двузначных чисел. Для каждого найденного числа запишите в ответе само число, наименьший и наибольший из его двузначных делителей. Так, например, для числа 36 учитываются только делители 12 и 18.

-
- 100) (Е. Джобс) Среди целых чисел, принадлежащих числовому отрезку [326496; 649632], найдите числа, у которых количество четных делителей равно количеству нечетных делителей. При этом в каждой из таких групп делителей не менее 70 элементов. Для каждого найденного числа запишите само число и минимальный делитель, больший 1000.
- 101) Среди целых чисел, принадлежащих числовому отрезку [125697; 190234], найдите числа, которые представляют собой произведение двух различных простых делителей. Запишите в ответе количество таких чисел и максимальное из них.
- 102) Среди целых чисел, принадлежащих числовому отрезку [268312; 336492], найдите числа, которые представляют собой произведение двух различных простых делителей. Запишите в ответе количество таких чисел и минимальное из них.
- 103) Среди целых чисел, принадлежащих числовому отрезку [351627; 428763], найдите числа, которые представляют собой произведение двух различных простых делителей. Запишите в ответе количество таких чисел и их среднее арифметическое. Для среднего арифметического запишите только целую часть числа.
- 104) Среди целых чисел, принадлежащих числовому отрезку [412567; 473265], найдите числа, которые представляют собой произведение двух различных простых делителей. Запишите в ответе количество таких чисел и то из них, которое ближе всего к их среднему арифметическому.
- 105) Рассматриваются целые числа, принадлежащих числовому отрезку [523456; 578925], которые представляют собой произведение двух различных простых делителей. Найдите такое из этих чисел, у которого два простых делителя меньше всего отличаются друг от друга. В ответе запишите простые делители этого числа в порядке возрастания. Если подходящих чисел несколько, запишите в ответе делители наименьшего из них.
- 106) Рассматриваются целые числа, принадлежащих числовому отрезку [631632; 684934], которые представляют собой произведение двух различных простых делителей. Найдите такое из этих чисел, у которого два простых делителя больше всего отличаются друг от друга. В ответе запишите простые делители этого числа в порядке возрастания. Если подходящих чисел несколько, запишите в ответе делители наименьшего из них.
- 107) Рассматриваются целые числа, принадлежащих числовому отрезку [153732; 225674], которые представляют собой произведение двух различных простых делителей. В ответе запишите количество таких чисел и такое из них, простые делители которого отличаются друг от друга меньше всего. Если чисел с наименьшей разностью делителей несколько, запишите в ответе наименьшее из них.
- 108) Рассматриваются целые числа, принадлежащих числовому отрезку [238941; 315675], которые представляют собой произведение двух различных простых делителей. В ответе запишите количество таких чисел и такое из них, простые делители которого отличаются друг от друга больше всего. Если чисел с наибольшей разностью делителей несколько, запишите в ответе наименьшее из них.
- 109) Среди целых чисел, принадлежащих числовому отрезку [173225; 217437], найдите числа, которые представляют собой произведение двух различных простых делителей, заканчивающихся на одну и ту же цифру. Запишите в ответе количество таких чисел и минимальное из них.
- 110) Среди целых чисел, принадлежащих числовому отрезку [237981; 309876], найдите числа, которые представляют собой произведение двух различных простых делителей, заканчивающихся на одну и ту же цифру. Запишите в ответе количество таких чисел и максимальное из них.
- 111) Среди целых чисел, принадлежащих числовому отрезку [264871; 322989], найдите числа, которые представляют собой произведение двух различных простых делителей, заканчивающихся на одну и ту же цифру. Запишите в ответе количество таких чисел и их среднее арифметическое. Для среднего арифметического запишите только целую часть числа.
-

- 112) Среди целых чисел, принадлежащих числовому отрезку [298435; 363249], найдите числа, которые представляют собой произведение двух различных простых делителей. Запишите в ответе количество таких чисел и то из них, которое ближе всего к их среднему арифметическому.
- 113) Рассматриваются целые числа, принадлежащих числовому отрезку [309829; 365874], которые представляют собой произведение двух различных простых делителей. Найдите такое из этих чисел, у которого два простых делителя меньше всего отличаются друг от друга. В ответе запишите простые делители этого числа в порядке возрастания. Если подходящих чисел несколько, запишите в ответе делители наименьшего из них.
- 114) Рассматриваются целые числа, принадлежащих числовому отрезку [326359, 421986], которые представляют собой произведение двух различных простых делителей. Найдите такое из этих чисел, у которого два простых делителя больше всего отличаются друг от друга. В ответе запишите простые делители этого числа в порядке возрастания. Если подходящих чисел несколько, запишите в ответе делители наименьшего из них.
- 115) Рассматриваются целые числа, принадлежащих числовому отрезку [478392; 502439], которые представляют собой произведение двух различных простых делителей. В ответе запишите количество таких чисел и такое из них, простые делители которого отличаются друг от друга меньше всего. Если чисел с наименьшей разностью делителей несколько, запишите в ответе наименьшее из них.
- 116) Рассматриваются целые числа, принадлежащих числовому отрезку [356738; 404321], которые представляют собой произведение двух различных простых делителей. В ответе запишите количество таких чисел и такое из них, простые делители которого отличаются друг от друга больше всего. Если чисел с наибольшей разностью делителей несколько, запишите в ответе наименьшее из них.
- 117) Рассматриваются целые числа, принадлежащих числовому отрезку [105673; 220784], которые представляют собой произведение трёх различных простых делителей. В ответе запишите количество таких чисел и максимальное из них.
- 118) Рассматриваются целые числа, принадлежащих числовому отрезку [158928; 345293], которые представляют собой произведение трёх различных простых делителей. В ответе запишите количество таких чисел и минимальное из них.
- 119) Рассматриваются целые числа, принадлежащих числовому отрезку [236228; 305283], которые представляют собой произведение трёх различных простых делителей. В ответе запишите количество таких чисел и их среднее арифметическое (только целую часть числа).
- 120) Рассматриваются целые числа, принадлежащих числовому отрезку [278932; 325396], которые представляют собой произведение трёх различных простых делителей, оканчивающихся на одну и ту же цифру. В ответе запишите количество таких чисел и максимальное из них.
- 121) Рассматриваются целые числа, принадлежащих числовому отрезку [318216; 369453], которые представляют собой произведение трёх различных простых делителей, оканчивающихся на одну и ту же цифру. В ответе запишите количество таких чисел и минимальное из них.
- 122) Рассматриваются целые числа, принадлежащих числовому отрезку [356712; 420901], которые представляют собой произведение трёх различных простых делителей, оканчивающихся на одну и ту же цифру. В ответе запишите количество таких чисел и их среднее арифметическое (только целую часть числа).
- 123) Рассматриваются целые числа, принадлежащих числовому отрезку [416782; 498324], которые представляют собой произведение трёх различных простых делителей, оканчивающихся на одну и ту же цифру. В ответе запишите количество таких чисел и разницу между максимальным и минимальным из них.

- 124) Рассматриваются целые числа, принадлежащих числовому отрезку [536792; 604298], которые представляют собой произведение трёх различных простых делителей, оканчивающихся на одну и ту же цифру. В ответе запишите количество таких чисел и такое из них, для которого разность наибольшего и наименьшего простых делителей максимальна.
- 125) Рассматриваются целые числа, принадлежащих числовому отрезку [485617; 529678], которые представляют собой произведение трёх различных простых делителей, оканчивающихся на одну и ту же цифру. В ответе запишите количество таких чисел и такое из них, для которого разность наибольшего и наименьшего простых делителей минимальна.
- 126) 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.
- 127) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку [1523467; 4157812] 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.
- 128) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку [4234679; 10157812] 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.
- 129) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку [12034679; 23175821] 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.
- 130) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку [50034679; 92136895] 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.
- 131) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку [106732567; 152673836] 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.
- 132) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку [247264322; 369757523] 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.
- 133) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку [358633892; 535672891] 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.
- 134) Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку $[525784203; 728943762]$ 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.
- 135) (**Е. Джобс**) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[321654; 654321]$, числа у которых есть только нечетные делители, количество которых больше 70. Делители 1 и само число не учитываются. Для каждого найденного числа запишите само число и максимальный по величине делитель.
- 136) (**Е. Джобс**) Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[25317; 51237]$, которые имеют хотя бы 6 различных простых делителей. Делители 1 и само число не учитываются. Запишите в ответе для каждого найденного числа само число и его максимальный простой делитель.
- 137) Рассмотрим произвольное натуральное число, представим его всеми возможными способами в виде произведения двух натуральных чисел и найдём для каждого такого произведения разность сомножителей. Например, для числа 18 получим: $18 = 18 \cdot 1 = 9 \cdot 2 = 6 \cdot 3$, множество разностей содержит числа 17, 7 и 3. Подходящей будем называть пару сомножителей, разность между которыми не превышает 90. Найдите все натуральные числа, принадлежащие отрезку $[500000; 1000000]$, у которых есть не менее трёх подходящих пар сомножителей. В ответе перечислите найденные числа в порядке возрастания, справа от каждого запишите наибольший из всех сомножителей, образующих подходящие пары.
- 138) Рассмотрим произвольное натуральное число, представим его всеми возможными способами в виде произведения двух натуральных чисел и найдём для каждого такого произведения разность сомножителей. Например, для числа 18 получим: $18 = 18 \cdot 1 = 9 \cdot 2 = 6 \cdot 3$, множество разностей содержит числа 17, 7 и 3. Подходящей будем называть пару сомножителей, разность между которыми не превышает 110. Найдите все натуральные числа, принадлежащие отрезку $[1000000; 1500000]$, у которых есть не менее трёх подходящих пар сомножителей. В ответе перечислите найденные числа в порядке возрастания, справа от каждого запишите наибольший из всех сомножителей, образующих подходящие пары.
- 139) Рассмотрим произвольное натуральное число, представим его всеми возможными способами в виде произведения двух натуральных чисел и найдём для каждого такого произведения разность сомножителей. Например, для числа 18 получим: $18 = 18 \cdot 1 = 9 \cdot 2 = 6 \cdot 3$, множество разностей содержит числа 17, 7 и 3. Подходящей будем называть пару сомножителей, разность между которыми не превышает 120. Найдите все натуральные числа, принадлежащие отрезку $[2000000; 3000000]$, у которых есть не менее трёх подходящих пар сомножителей. В ответе перечислите найденные числа в порядке возрастания, справа от каждого запишите наибольший из всех сомножителей, образующих подходящие пары.
- 140) (**А. Рулин**) Рассматривается множество целых чисел, принадлежащих числовому отрезку $[854321; 1087654]$. Найдите числа, нетривиальные делители которых образуют арифметическую прогрессию с разностью $d = 10$. В ответе для каждого такого числа (в порядке возрастания) запишите сначала само число, а потом – его минимальный нетривиальный делитель.
- 141) (**А. Рулин**) Рассматривается множество целых чисел, принадлежащих числовому отрезку $[834567; 1143210]$. Найдите числа, нетривиальные делители которых образуют арифметическую прогрессию с разностью $d = 2$. В ответе для каждого такого числа (в порядке возрастания) запишите сначала само число, а потом – его максимальный нетривиальный делитель.

- 142) (**А. Рулин**) Рассматривается множество целых чисел, принадлежащих числовому отрезку [862346; 1056242]. Найдите числа, нетривиальные делители которых образуют арифметическую прогрессию с разностью $d = 100$. В ответе для каждого такого числа (в порядке возрастания) запишите сначала само число, а потом – его максимальный нетривиальный делитель.
- 143) (**Е. Джобс**) Для интервала [33333;55555] найти все простые числа, сумма цифр которых больше 35. Запишите найденные числа в порядке возрастания, справа от каждого – сумму его цифр.
- 144) (**Е. Джобс**) Для интервала [33333;55555] найдите числа, которые кратны сумме своих простых собственных делителей (меньших самого числа). В качестве ответа приведите в порядке возрастания числа, для которых сумма простых делителей больше 250, после каждого числа запишите сумму его простых собственных делителей.
- 145) (**С.О. Куров**) Среди целых чисел, принадлежащих числовому отрезку [1000000; 1300000], найдите числа, у которых все цифры меньше тройки, а сумма цифр кратна десяти. Из всех таких чисел необходимо отобрать 10-е, 20-е, 30-е и так далее. Расположите найденные числа в порядке возрастания, справа от каждого числа укажите количество его собственных делителей (не равных 1 и самому числу).
- 146) Найдите все натуральные числа, принадлежащие отрезку [100 000 000; 101 000 000], у которых ровно три различных чётных делителя. В ответе перечислите найденные числа в порядке возрастания, справа от каждого числа запишите его второй по величине нетривиальный делитель (не равный 1 и самому числу).
- 147) Найдите все натуральные числа, принадлежащие отрезку [103 000 000; 104 000 000], у которых ровно три различных чётных делителя. В ответе перечислите найденные числа в порядке возрастания, справа от каждого числа запишите его второй по величине нетривиальный делитель (не равный 1 и самому числу).
- 148) Найдите все натуральные числа, принадлежащие отрезку [113 000 000; 114 000 000], у которых ровно три различных чётных делителя. В ответе перечислите найденные числа в порядке возрастания, справа от каждого числа запишите его второй по величине нетривиальный делитель (не равный 1 и самому числу).
- 149) Найдите все натуральные числа, принадлежащие отрезку [55 000 000; 60 000 000], у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа, справа от каждого числа запишите его наибольший нечётный делитель.
- 150) Найдите все натуральные числа, принадлежащие отрезку [105 000 000; 115 000 000], у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа, справа от каждого числа запишите его наибольший нечётный делитель.
- 151) Найдите все натуральные числа, принадлежащие отрезку [78 000 000; 85 000 000], у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа, справа от каждого числа запишите его наибольший нечётный делитель.
- 152) Найдите все натуральные числа, принадлежащие отрезку [63 000 000; 75 000 000], у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа, справа от каждого числа запишите его наибольший нечётный делитель.
- 153) (**А. Богданов**) Найдите наименьшее натуральное число, которое имеет ровно 1600 делителей. В ответе запишите сначала само число и затем его наибольший простой делитель. Подсказка: используйте основную теорему арифметики.

- 154) **(А. Богданов)** Найдите наименьшее натуральное число, которое имеет ровно 1200 делителей. В ответе запишите сначала само число и затем его наибольший простой делитель. Подсказка: используйте основную теорему арифметики.
- 155) **(А. Богданов)** Найдите наименьшее натуральное число, которое имеет ровно 1000 делителей. В ответе запишите сначала само число и затем его наибольший простой делитель. Подсказка: используйте основную теорему арифметики.
- 156) **(А. Богданов)** Найдите наименьшее натуральное число, которое имеет ровно 729 делителей. В ответе запишите сначала само число и затем его наибольший простой делитель. Подсказка: используйте основную теорему арифметики.
- 157) **(А. Богданов)** Найдите наименьшее натуральное число, которое имеет ровно 512 делителей. В ответе запишите сначала само число и затем его наибольший простой делитель. Подсказка: используйте основную теорему арифметики.
- 158) **(Е. Джобс)** Найдите возрастающую последовательность из 5 чисел, начинающуюся с 700000, такую, что каждый следующий элемент – это минимальное число, количество делителей которого превосходит количество делителей предыдущего числа. Для каждого элемента последовательности запишите сначала само число, а затем количество его натуральных делителей.
- 159) Рассматриваются возрастающие последовательности из 5 **идущих подряд** чисел, больших 700000, такие, что количество делителей каждого следующего числа превосходит количество делителей предыдущего числа. Найдите такую последовательность, которая начинается с наименьшего возможного числа. Для каждого числа из этой последовательности запишите сначала само число, а затем количество его натуральных делителей.
- 160) **(Е. Джобс)** Напишите программу, которая находит 6 простых чисел наиболее приближенные к числу 10000000 (10 миллионов). Причем 3 найденных числа должны быть меньше 10000000, остальные 3 числа – больше. Найденные числа расположите в порядке возрастания. В качестве ответа выведите пары чисел – расстояние от найденного числа до 10000000 и само число.
- 161) Найдите все натуральные числа, N , принадлежащие отрезку $[150\,000\,000; 300\,000\,000]$, которые можно представить в виде $N = 2^m \cdot 3^n$, где m – чётное число, n – нечётное число. В ответе запишите все найденные числа в порядке возрастания, а справа от каждого числа – сумму $m+n$.
- 162) Найдите все натуральные числа, N , принадлежащие отрезку $[150\,000\,000; 300\,000\,000]$, которые можно представить в виде $N = 2^m \cdot 3^n$, где m – нечётное число, n – чётное число. В ответе запишите все найденные числа в порядке возрастания, а справа от каждого числа – сумму $m+n$.
- 163) Найдите все натуральные числа, N , принадлежащие отрезку $[100\,000\,000; 300\,000\,000]$, которые можно представить в виде $N = 2^m \cdot 5^n$, где m – чётное число, n – нечётное число. В ответе запишите все найденные числа в порядке возрастания, а справа от каждого числа – сумму $m+n$.
- 164) Найдите все натуральные числа, N , принадлежащие отрезку $[100\,000\,000; 300\,000\,000]$, которые можно представить в виде $N = 2^m \cdot 5^n$, где m – нечётное число, n – чётное число. В ответе запишите все найденные числа в порядке возрастания, а справа от каждого числа – сумму $m+n$.
- 165) Найдите все натуральные числа, N , принадлежащие отрезку $[100\,000\,000; 300\,000\,000]$, которые можно представить в виде $N = 2^m \cdot 7^n$, где m – чётное число, n – нечётное число. В ответе запишите все найденные числа в порядке возрастания, а справа от каждого числа – сумму $m+n$.
- 166) Найдите все натуральные числа, N , принадлежащие отрезку $[100\,000\,000; 300\,000\,000]$, которые можно представить в виде $N = 2^m \cdot 7^n$, где m – нечётное число, n – чётное число. В ответе запишите все найденные числа в порядке возрастания, а справа от каждого числа – сумму $m+n$.
- 167) **(Н. Плотцын)** Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[3; 1000000]$ последовательности подряд идущих составных чисел длиной не менее 90. Для каждой найденной последовательности запишите в порядке возрастания простые

числа, стоящие на границах данных последовательностей. В ответе запишите эти пары простых чисел в порядке возрастания первого числа в паре.

- 168) **(А. Кабанов)** Обозначим через S сумму всех натуральных делителей целого числа, кроме единицы и самого числа. Если таких делителей у числа нет, то считаем значение S равным нулю. Напишите программу, которая перебирает целые числа, большие 150000 в порядке возрастания и ищет среди них такие, для которых значение S при делении на 13 даёт остаток 10. Программа должна найти и первые 7 таких чисел. Для каждого из них запишите в отдельной строке сначала само число, затем значение S . Строки выводятся в порядке возрастания найденных чисел.
- 169) **(А. Кабанов)** Обозначим через S сумму **простых** делителей целого числа, не считая самого числа. Если таких делителей у числа нет, то считаем значение S равным нулю. Напишите программу, которая перебирает целые числа, большие 250000 в порядке возрастания и ищет среди них такие, для которых значение S не равно нулю и кратно 17. Программа должна найти первые 5 таких чисел. Для каждого из них в отдельной строке сначала выводится само число, затем значение S . Строки выводятся в порядке возрастания найденных чисел.
- 170) **(А. Кабанов)** Обозначим через M разность максимального и минимального натуральных делителей целого числа, не считая единицы и самого числа. Если таких делителей у числа нет, то считаем значение M равным нулю. Напишите программу, которая перебирает целые числа, большие 350000, в порядке возрастания и ищет среди них такие, для которых значение M при делении на 23 даёт в остатке 9. Запишите первые 6 найденных чисел в порядке возрастания, справа от каждого числа запишите соответствующее значение M .
- 171) **(А. Кабанов)** Обозначим через M разность максимального и минимального числа среди **простых** делителей целого числа, не считая самого числа. Если таких делителей у числа нет, то считаем значение M равным нулю. Напишите программу, которая перебирает целые числа, большие 450000, в порядке возрастания и ищет среди них такие, для которых значение M при делении на 29 даёт в остатке 11. Выведите первые 4 найденных числа в порядке возрастания, справа от каждого числа запишите соответствующее значения M .
- 172) **(А. Кабанов)** Обозначим через F целую часть среднего арифметического всех натуральных делителей целого числа, кроме единицы и самого числа. Если таких делителей у числа нет, то считаем значение F равным нулю. Напишите программу, которая перебирает целые числа, большие 550000, в порядке возрастания и ищет среди них такие, для которых значение F при делении на 31 даёт в остатке 13. Выведите первые 5 найденных чисел в порядке возрастания и справа от каждого числа – соответствующее значение F .
- 173) **(А. Кабанов)** Обозначим через F целую часть среднего арифметического всех **простых** делителей целого числа, не считая самого числа. Если таких делителей у числа нет, то считаем значение F равным нулю. Напишите программу, которая перебирает целые числа, большие 650000, в порядке возрастания и ищет среди них такие, для которых значение F при делении на 37 даёт в остатке 23. Выведите первые 4 найденных числа в порядке возрастания и справа от каждого числа – соответствующее значение F .
- 174) **(С. Неретин)** Пифагоровой тройкой назовём тройку чисел (a, b, c) , такую что $a \leq b \leq c$ и $a^2 + b^2 = c^2$. Найдите все пифагоровы тройки, в которых все числа находятся в диапазоне $[1; 5000]$. Запишите в ответе количество подходящих троек, а затем – значение c для тройки, в которой сумма $a+b+c$ максимальна.
- 175) **(Б. Баобаба)** Числа-близнецы — это такие простые числа, которые отличаются друг от друга на 2. Найдите все пары чисел-близнецов в диапазоне $[3\ 000\ 000; 10\ 000\ 000]$. В ответе запишите количество найденных пар и среднее арифметическое последней пары.
- 176) **(А. Комков)** Пусть A – абсолютное значение разности максимального четного и максимального нечетного делителей числа, не считая единицы и самого числа. Если хотя бы одного из таких

делителей у числа нет, то считаем значение A равным нулю. Напишите программу, которая перебирает целые числа, большие 250156, в порядке возрастания и ищет среди них первые 5, для которых значение A является простым числом, оканчивающимся на 9. Для каждого из найденных чисел в отдельной строке сначала выводить само число, затем значение A . Строки выводятся в порядке возрастания найденных чисел.

- 177) (**А. Комков**) Обозначим через S сумму делителей числа, не являющихся простыми, кроме единицы и самого числа. Если таких делителей у числа нет, то S равно нулю. Напишите программу, которая перебирает нечетные целые числа, меньшие 912673, в порядке убывания и ищет среди них первые 5 чисел, которые кратны S . Для каждого из найденных чисел в отдельной строке сначала выводится само число, затем значение S . Строки выводятся в порядке убывания найденных чисел.
- 178) 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.
- 179) Найдите 5 чисел больших 800000, таких, что сумма их наименьшего и наибольшего нетривиальных делителей (не считая единицы и самого числа) делится на 138. В качестве ответа приведите 5 наименьших составных (не простых) чисел, соответствующих условию. Формат вывода: для каждого из найденных чисел в отдельной строке запишите само число, а затем сумму его наименьшего и наибольшего нетривиальных делителей.
- 180) (**Л. Шагин**) Среди чисел, больших 520000, найти такие, сумма всех делителей которых, не считая единицы и самого числа, образует число-палиндром (например, число 1221: если его «перевернуть», получается то же самое число). Вывести первые пять чисел, удовлетворяющих вышеописанному условию, справа от каждого числа вывести его максимальный делитель.
- 181) (**А. Богданов**) Среди чисел, больших куба максимального простого двузначного числа, найдите 5 минимальных чисел, у которых есть ровно три различных трехзначных делителя, оканчивающихся на 3. Для каждого из 5 найденных чисел выводится само число, а затем его минимальный трехзначный делитель, оканчивающийся на 3.
- 182) (**Л. Шагин**) Последовательность Люка – это последовательность чисел, в которых каждое последующее число образуется из суммы двух предшествующих ему чисел. Первые два числа в последовательности Люка: 2, 1. Найдите все простые числа Люка, принадлежащие отрезку $[10^6; 10^9]$. Для каждого найденного числа выведите сначала номер числа в последовательности Люка, а затем само число.
- 183) 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 $P(N)$.
- 184) 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 300 000 000, для которых $P(N)$ оканчивается на 31 и не превышает N . В ответе для каждого найденного числа запишите сначала значение $P(N)$, а затем – наибольший делитель, вошедший в произведение $P(N)$.

- 185) 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 400 000 000, для которых $P(N)$ оканчивается на 17 и не превышает N . В ответе для каждого найденного числа запишите сначала значение $P(N)$, а затем – наибольший делитель, вошедший в произведение $P(N)$.
- 186) 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 500 000 000, для которых $P(N)$ оканчивается на 91 и не превышает N . В ответе для каждого найденного числа запишите сначала значение $P(N)$, а затем – наибольший делитель, вошедший в произведение $P(N)$.
- 187) Пусть $S(N)$ – сумма двух наибольших нетривиальных делителей числа N (не считая единицы и самого числа). Если у числа N меньше двух таких делителей, то $S(N)$ считается равным 0. Найдите 5 наименьших натуральных чисел, превышающих 10 000 000, для которых $S(N)$ меньше, чем 100 000, и кратно 31. В ответе запишите найденные значения $S(N)$ в порядке возрастания соответствующих им чисел N .
- 188) Пусть $S(N)$ – сумма двух наибольших нетривиальных делителей числа N (не считая единицы и самого числа). Если у числа N меньше двух таких делителей, то $S(N)$ считается равным 0. Найдите 5 наименьших натуральных чисел, превышающих 10 000 000, для которых $S(N)$ меньше, чем 100 000, и десятичная запись этого числа оканчивается на 112. В ответе запишите найденные значения $S(N)$ в порядке возрастания соответствующих им чисел N .
- 189) 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 $S(N)$ – простое число. В ответе запишите найденные значения $S(N)$ в порядке возрастания соответствующих им чисел N .
- 190) 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 значения $S(N)$ в порядке возрастания соответствующих им чисел N .
- 191) 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 десятичная запись $S(N)$ содержит не менее 4-х цифр 7. В ответе запишите найденные значения $S(N)$ в порядке возрастания соответствующих им чисел N .
- 192) 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 в десятичной записи $S(N)$ все цифры расположены в порядке неубывания. В ответе запишите найденные значения $S(N)$ в порядке возрастания соответствующих им чисел N .