

Задание №25

Обработка целых чисел.

Проверка делимости.

Время выполнения 20 минут

Про делители.

1. У каждого числа n есть два тривиальных делителя единица и само число (1 и n).
2. Нетривиальные делители числа:
 - наибольший нетривиальный делитель натурального числа - это наибольшее натуральное число на которое делится число
 - наибольший нетривиальный делитель по порядку записи в массиве стоит предпоследним.

Поиск делителей

Поиск делителей числа и разложение его на множители - разные понятия, хотя термины в какой-то степени схожи.

Под делителями числа подразумевают все числа, на которые оно делится.

Например для числа 20:

Множители 2,2,5

Делители 1,2,4,5,10,20

Про числа.

1. Множество натуральных чисел делится на простые и составные числа.
2. В основе этой классификации лежит понятие делимости натуральных чисел.
3. Числа, у которых есть нетривиальные делители называются составными.
4. Число n называется простым, если у него только тривиальные делители, т.е. 1 и n .

Про задачи.

1. Во всех задачах осуществляется перебор целых чисел на определённом отрезке и

Например, надо подсчитать количество чисел, удовлетворяющих какому-то условию.

Цикл в общем виде

Pascal	Python	C++
<pre>count := 0; for n:=a to b do if <i>условие выполнено</i> then count := count + 1; writeln(count);</pre>	<pre>count = 0 for n in range(a, b+1): if <i>условие выполнено</i>: count += 1 print(count)</pre>	<pre>int count = 0; for(int n = a; n <= b; n++) if(<i>условие выполнено</i>) count += 1; std::cout << count;</pre>

Проверка делимости

Pascal	Python	C++
<pre>if n mod d = 0 then writeln('Делится') else writeln('Не делится')</pre>	<pre>if n % d == 0: print("Делится") else: print("Не делится")</pre>	<pre>if(n % d == 0) std::cout << "Делится"; else std::cout << "Не делится";</pre>

Количество делителей

Pascal	Python	C++
<pre>count := 0; for d:=1 to n do if n mod d = 0 then count := count + 1; writeln(count);</pre>	<pre>count = 0 for d in range(1, n+1): if n % d == 0: count += 1 print(count)</pre>	<pre>int count = 0; for(int d = 1; d <= n; d++) if(n % d == 0) count ++; std::cout << count;</pre>

В задачах требуется определить не только количество делителей, но и сами делители, тогда их нужно сохранять в списке (массиве).

В **Python'e** удобно использовать динамический список (массив): сначала он пустой, если нашли делитель, то добавили в список (массив):

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

Pascal и C++

Проще без динамического массива! Как ?

1. Выделяем массив для хранения всех делителей;

(количество делителей числа n не больше самого числа!!!).

ИЛИ

2. Если по условию задачи нужны числа, имеющие 4 делителя (или 5; 6 делителей),

то выделяем массив из 4-х элементов (или из 5; 6 элементов),

а остальные делители в массив не записываем.

Перебор делителей можно оптимизировать, учитывая, что наименьший из пары делителей, таких что $a * b = n$, не превышает квадратного корня из n ; нужно только аккуратно обработать случай, когда число n представляет собой квадрат другого целого числа;

Пример № 1.

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [126849; 126871], числа, **имеющие ровно 4 различных делителя.**

Выведите эти четыре делителя для каждого найденного числа в порядке возрастания.

Пример №1. Программа Python

```
for n in range(126849, 126871+1):  
  
    a = []                #список делителей пустой  
  
    for d in range(1, n+1):    # перебираем все делители  
  
        if n % d == 0:        # если нашли  
  
            a.append(d)        # добавляем в список  
  
            if len(a) > 4: break  
  
    if len(a) == 4:  
        print(*a)
```

Пример №1. Программа Python

```
1 for n in range(126849, 126871+1):
2     a = [] #список делителей
3     for d in range(1,n+1): # перебираем все делители
4         if n % d == 0: # если нашли
5             a.append(d) # добавляем в список
6             if len(a) > 4: break
7     if len(a) == 4:
8         print(*a)
```

Отве

т:

```
1 3 42283 126849
1 47 2699 126853
1 5 25373 126865
1 293 433 126869
```

Пример №1. Программа Pascal

```
var
    numDel, i, j: longint;
    d: array[1..4] of longint;
begin
    for i := 126849 to 126871 do begin
        numDel := 0;
        for j := 1 to (i + 1) do begin
            if i mod j = 0 then begin
                numDel := numDel + 1;
                if numDel > 4 then break;
                d[numDel] := j;
            end;
        end;
        if numDel = 4 then writeln(d[1], ' ', d[2], ' ', d[3], ' ', d[4]);
    end;
end.
```

Пример №1. Программа Pascal

```
1 var
2   numDel, i, j: longint;
3   d: array[1..4] of longint;
4 begin
5   for i := 126849 to 126871 do begin
6     numDel := 0;
7     for j := 1 to (i + 1) do begin
8       if i mod j = 0 then begin
9         numDel := numDel + 1;
10        if numDel > 4 then break;
11        d[numDel] := j;
12      end;
13    end;
14    if numDel = 4 then writeln(d[1], ' ', d[2], ' ', d[3], ' ', d[4]);
15  end;
16 end.
```

ОТВЕТ:

```
1 3 42283 126849
1 47 2699 126853
1 5 25373 126865
1 293 433 126869
```

Пример №

2.

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[210\ 235; 210\ 300]$, числа, имеющие ровно четыре различных натуральных делителя, **не считая единицы и самого числа**.

Для каждого найденного числа запишите эти четыре делителя в четыре соседних столбца на экране с новой строки.

Делители в строке должны следовать в порядке возрастания.

Например, в диапазоне $[10; 16]$ ровно четыре различных натуральных делителя имеет число 12, поэтому для этого диапазона вывод на экране должна содержать следующие значения: 2 3 4 6

Пример №2. Программа Python

```
for n in range(210235, 210300+1):  
    a = []  
    for d in range(2, n//2 + 1): # диапазон!!!  
        if n % d == 0:  
            a.append(d)  
            if len(a) > 4: break  
    if len(a) == 4:  
        print(*a)
```

```
1 ▾ for n in range(210235, 210300+1):  
2     a = []  
3 ▾     for d in range(2, n//2 + 1):  
4 ▾         if n % d == 0:  
5             a.append(d)  
6             if len(a) > 4: break  
7 ▾     if len(a) == 4:  
8         print(*a)
```

Ответ:

```
2 4 52561 105122  
2 4 52567 105134  
2 4 52571 105142
```

Пример №2. Программа Python

```
**from math import sqrt
```

```
for n in range(210235, 210300+1):
```

```
    a = []
```

```
    q= int(sqrt(n))
```

```
    for d in range(2, q +1):
```

```
        if n % d == 0:
```

```
            if d==n//d: # если делитель является точным квадратом числа,
```

```
                a=a+[d]    # то добавляем один делитель
```

```
            else:
```

```
                a=a+[ d, n//d] # если нет- то добавляем 2 делителя
```

```
            if len(a) > 4: break
```

```
    if len(a) == 4:
```

```
        print(*a)
```

**** *импортируем модуль math сразу же с названием функции sqrt***

если импортировать только модуль math без названия функции,
то тогда в коде надо писать math.sqrt

```
2 105122 4 52561
2 105134 4 52567
2 105142 4 52571
```

Пример №2. Программа Pascal

```
1  const divCount = 4;
2  var n, count, d, i, j, q: longint;
3      divs: array[1..divCount] of longint;
4  begin
5      for n:=210235 to 210300 do begin
6          count := 0;
7          q := round(sqrt(n));
8          for d:=2 to q do
9              if n mod d = 0 then begin
10                 count := count + 2;
11                 if count <= divCount then begin
12                     divs[count-1] := d;
13                     if d <> n div d then
14                         divs[count] := n div d;
15                 end
16                 else break
17             end;
18         if count = divCount then begin
19             for i:=1 to divCount do
20                 write(divs[i], ' ');
21             writeln
22         end
23     end
24 end.
```

```
const divCount = 4;
var n, count, d, i, j, q: longint;
    divs: array[1..divCount] of longint;
begin
    for n:=210235 to 210300 do begin
        count := 0;
        q := round(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
                else break
            end;
        if count = divCount then begin
            for i:=1 to divCount do
                write(divs[i], ' ');
            writeln
        end
    end
end
end.
```

Пример № 3.

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[110203, 110245]$, число, **имеющее максимальное количество различных натуральных делителей**, если таких чисел несколько — **найдите минимальное из них**. Выведите на экран количество делителей такого числа и само число.

Например, в диапазоне $[2; 48]$ максимальное количество различных натуральных делителей имеет число 48, поэтому для этого диапазона вывод на экране должна содержать следующие значения: 10 48

Пример №3. Программа Python

```
Nmin, Maxc = 0,0          #минимальное число, счётчик делителей
for N in range (110203, 110245 + 1):
    count=0
    for d in range (1, N + 1):
        if N % d == 0:
            count+= 1
    if count > Maxc:
        Maxc = count
        Nmin = N

print (Maxc, Nmin)
```

Отве

64 110208

т:

Пример №3. Программа Python

```
1 Nmin, Maxc = 0,0
2 for N in range (110203, 110245 + 1):
3     count=0
4     for d in range (1, N + 1):
5         if N % d == 0:
6             count+= 1
7     if count > Maxc:
8         Maxc = count
9         Nmin = N
10
11 print (Maxc, Nmin)
```

Пример №3. Программа Pascal

```
var
    minN, numDel, maxDel, N, del: longint;
begin
    minN := 0;
    maxDel := 0;
    for N := 110203 to 110245 do begin
        numDel := 0;
        for del := 1 to N do begin
            if (N mod del = 0) then begin
                numDel := numDel + 1;
            end;
        end;
        if (numDel > maxDel) then begin
            maxDel := numDel;
            minN := N;
        end;
    end;
    writeln(maxDel, ' ', minN);
end.
```

ОТВЕТ: 64 110208

Пример №4.

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [190201; 190230], числа, имеющие ровно 4 различных делителя.

Выведите эти четыре делителя для каждого найденного числа **в порядке убывания.**

Пример №4. Программа Python

```
for n in range(190201,190230+1):
    a = []
    for d in range(1,n+1):
        if n % d == 0:
            a.append(d)
            if len(a) > 4: break
    if len(a) == 4:
        a.reverse()      # вывод в порядке убывания
        print(*a)
```

```
1 for n in range(190201,190230+1):
2     a = []
3     for d in range(1,n+1):
4         if n % d == 0:
5             a.append(d)
6             if len(a) > 4: break
7     if len(a) == 4:
8         a.reverse()
9         print(*a)
```

Ответ

:

```
190201 17291 11 1
190202 95101 2 1
190214 95107 2 1
190219 853 223 1
190222 95111 2 1
190223 17293 11 1
190227 63409 3 1
190229 14633 13 1
```

Пример №4. Программа Pascal

```
1 var
2   numDel, i, j: longint;
3   d: array[1..4] of longint;
4 begin
5   for i := 190201 to 190230 do begin
6     numDel := 0;
7     for j := 1 to (i + 1) do begin
8       if i mod j = 0 then begin
9         numDel := numDel + 1;
10        if numDel > 4 then break;
11        d[numDel] := j;
12      end;
13    end;
14    if numDel = 4 then writeln(d[4], ' ', d[3], ' ', d[2], ' ', d[1]);
15  end;
16 end.
```

Самостоятельно

1. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[190201; 190280]$, числа, имеющие ровно четыре различных чётных натуральных делителя. Для каждого найденного числа запишите эти четыре делителя в четыре соседних столбца на экране с новой строки. Делители в строке должны следовать в порядке убывания.
2. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[193136; 193223]$, числа, имеющие ровно 6 различных делителей. Выведите эти делители для каждого найденного числа в порядке возрастания.
3. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[258274; 258397]$, числа, имеющие ровно четыре различных натуральных делителя, не считая единицы и самого числа. Для каждого найденного числа запишите эти четыре делителя в четыре соседних столбца на экране с новой строки. Делители в строке должны следовать в порядке возрастания.

Решения и ответы.

№1

```
1 for n in range(190201,190280+1):
2     a = []
3     for d in range(1,n+1):
4         if n % d == 0 and d%2==0:
5             a.append(d)
6             if len(a) > 4: break
7     if len(a) == 4:
8         a.reverse()
9         print(*a)
```

```
190226 838 454 2
190234 17294 22 2
190238 2606 146 2
190252 95126 4 2
190258 758 502 2
190274 27182 14 2
190276 95138 4 2
```

№2

2

```
1 for n in range(193136, 193223+1):
2     a = []
3     for d in range(1,n+1):
4         if n % d == 0:
5             a.append(d)
6             if len(a) > 6: break
7     if len(a) == 6:
8         print(*a)
```

```
1 5 25 7727 38635 193175
1 2 4 48299 96598 193196
1 3 9 21467 64401 193203
1 7 49 3943 27601 193207
```

Решения и ответы.

№3

```
1 for n in range(258274, 258397+1):
2     a = []
3     for d in range(2, n//2 +1):
4         if n % d == 0:
5             a.append(d)
6             if len(a) > 4: break
7     if len(a) == 4:
8         |
9         print(*a)
```

```
5 25 10331 51655
2 4 64577 129154
2 4 64579 129158
5 25 10333 51665
3 9 28703 86109
2 4 64591 129182
7 49 5273 36911
```

Разные типы чисел

Простые числа

1. Простые числа – это натуральные числа, больше единицы, которые имеют только два делителя: единицу и само число.

(Единица не является простым числом!)

2. Следовательно, для определения простоты числа надо перебрать его делители, и, если их всего два, то это число простое.

3. Для ускорения перебора делителей числа нужно использовать свойство: наименьший из пары делителей, не превышает квадратного корня из n ;

т.е. интервал для перебора делителей

$[2, \sqrt{n}]$

Python:

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

Pascal:

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

C++:

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

Пример № 1.

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.

Пример №1. Программа Python

```
1 k=0          # порядковый номер
2
3 for n in range(2943444, 2943529+1):
4     count = 2      # есть уже делители: 1 и само число
5     for d in range(2, round(n**0.5)+1): # поиск других делителей
6         if n % d == 0:
7             count = 0
8             break      # число уже не простое
9     if count == 2:
10        k +=1
11        print(k,n)
```

**Отве
т:**

```
1 2943467
2 2943491
3 2943503
4 2943527
```

Самостоятельно

1. 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.
2. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [5408238; 5408389], простые числа. Выведите все найденные простые числа в порядке возрастания, слева от каждого числа выведите его номер по порядку.
3. 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.
4. Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [2532000; 2532160], простые числа. Найдите все простые числа, которые заканчиваются на цифру 3. Выведите их в порядке возрастания, слева от каждого числа выведите его номер по порядку.

Ответ ы.

№1

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

№2

```
1 5408309  
2 5408311  
3 5408323  
4 5408341  
5 5408357  
6 5408383  
7 5408387
```

№ 3

```
1 2532007  
2 2532067  
3 2532071  
4 2532083  
5 2532107
```

№ 4

```
1 2532083  
2 2532113  
3 2532143
```

Решения.

№ 3

```
1 k=0
2 for n in range(2532000, 2532160+1):
3     count = 2
4     for d in range(2, round(n**0.5)+1):
5         if n % d == 0:
6             count = 0
7             break
8     if count == 2 and k < 5:
9         k +=1
10        print(k,n)
11        if k == 5: break
```

№ 4

```
1 k=0
2
3 for n in range(2532000, 2532160+1):
4     count = 2
5     for d in range(2, round(n**0.5)+1):
6         if n % d == 0:
7             count = 0
8             break
9     if count == 2 and n%10 == 3:
10        k +=1
11        print(k,n)
```

Совершенные числа

Совершенным называется число, натуральное число, равное сумме всех своих собственных делителей (то есть всех положительных делителей, отличных от самого числа) (например, число $6=1+2+3$).

Пример № 1.

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.

```
1 for n in range(2,10000+1):  
2     a = []  
3     for d in range(1,n//2+1):  
4         if n % d == 0:  
5             a.append(d)  
6     if n== sum(a):  
7         print(n,len(a))
```

Ответ:

```
6 3  
28 5  
496 9  
8128 13
```

Самостоятельно

1. Выведите каждое почти совершенное число из диапазона [1000; 10000] в порядке возрастания по одному в строке. Число называется почти совершенным, если оно меньше суммы своих собственных делителей (то есть всех положительных делителей, отличных от самого числа) на единицу.

Отве

т:

1024

2048

4096

8192