

Обработка целочисленной информации

Решаем задачи ЕГЭ № 25

[Курсы ISOT МГТУ им. Н. Э. Баумана](https://isot.bmstu.ru/)

<https://isot.bmstu.ru/>

Спецификация задач: задача №25

- Новые задачи 2021 года
- Умение создавать собственные программы для обработки целочисленной информации
- Содержание: Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности
- Умение: Строить информационные модели объектов, систем и процессов в виде алгоритмов
- Уровень сложности: высокий
- Примерное время выполнения: 35 минут

Решение задач № 25 (ДЕМО-2021)

25

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.

Например, в диапазоне $[5; 9]$ ровно два целых различных натуральных делителя имеют числа 6 и 8, поэтому для этого диапазона таблица на экране должна содержать следующие значения:

2	3
2	4

Ответ:

...	...

Решение через нахождение количества и печать: Python (без функций)

```
for number in range(174457, 174505+1):
```

```
    cnt = 0
```

```
    for i in range(2, number):
```

```
        if number % i == 0:
```

```
            cnt += 1
```

```
    if cnt == 2:
```

```
        for i in range(2, number):
```

```
            if number % i == 0:
```

```
                print(i, end = ' ')
```

```
    print()
```

Решение через нахождение количества и печать: Python (с функциями)

```
def divCount(number):  
    cnt = 0  
    for i in range(2, number):  
        if number % i == 0:  
            cnt += 1  
    return cnt
```

```
def divPrint(number):
```

Решение через нахождение количества и печать: C++ (без функций)

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    for(int number = 174457; number <= 174505; number++){  
        int cnt = 0;  
        for(int i = 2; i < number; i++){  
            if (number % i == 0)  
                cnt += 1;  
        }  
        if (cnt == 2){  
            for(int i = 2; i < number; i++){
```

Решение через нахождение количества и печать: C++ (с функциями)

```
#include <iostream>
using namespace std;
```

```
int divCount(int number){
    int cnt = 0;
    for(int i = 2; i < number; i++)
        if (number % i == 0)
            cnt += 1;
    return cnt;
}
```

```
void divPrint(int number){
```

Решение через нахождение количества и печать: Pascal (без функций)

```
var number, cnt, i: integer;
begin
  for number := 174457 to 174505 do
  begin
    cnt := 0;
    for i := 2 to number-1 do
      if number mod i = 0 then
        cnt := cnt + 1;

    if cnt = 2 then
      begin
```

Решение через нахождение количества и печать: Pascal (с функциями)

```
function divCount(number: integer): integer;
var cnt, i: integer;
begin
    cnt := 0;
    for i := 2 to number-1 do
        if number mod i = 0 then
            cnt := cnt + 1;
    divCount := cnt;
end;
```

```
procedure divPrint(number: integer);
var i: integer;
begin
    for i := 2 to number-1 do
```

Решение через динамические структуры данных: Python

```
def dividers(number):  
    d = []  
    for i in range(2, number):  
        if number % i == 0:  
            d.append(i)  
    return d  
  
for i in range(174457, 174505+1):  
    divs = dividers(i)  
    if (len(divs) == 2):  
        print(divs)
```

Решение через динамические структуры данных: C++

```
#include <iostream>
#include <vector>
using namespace std;

vector<int> dividers(int number){
    vector<int> d;
    for(int i = 2; i < number; i++){
        if (number % i == 0){
            d.push_back(i);
        }
    }
    return d;
}
```

Решение через динамические структуры данных: C++

```
#include <iostream>
#include <vector>
using namespace std;

vector<int> dividers(int number){
    vector<int> d;
    for(int i = 2; i < number; i++){
        if (number % i == 0){
            d.push_back(i);
        }
    }
    return d;
}
```

Решение через динамические структуры данных: Pascal

```
function dividers(number: integer): List<integer>;
var
  d: List<integer> := nil;
  i: integer;
begin
  d := new List<integer>;
  for i := 2 to number - 1 do
    if (number mod i = 0) then
      begin
        d := d + Lst(i);
      end;
  end;
  dividers := d;
end;
```

Задание демоварианта - 2022

25

Пусть M – сумма минимального и максимального натуральных делителей целого числа, не считая единицы и самого числа. Если таких делителей у числа нет, то значение M считается равным нулю.

Напишите программу, которая перебирает целые числа, бóльшие 700 000, в порядке возрастания и ищет среди них такие, для которых значение M оканчивается на 8. Выведите первые пять найденных чисел и соответствующие им значения M .

Формат вывода: для каждого из пяти таких найденных чисел в отдельной строке сначала выводится само число, затем – значение M .

Строки выводятся в порядке возрастания найденных чисел.

Количество строк в таблице для ответа избыточно.

Ответ:

...	...

Задание демоварианта - 2022

25	700005	233338
	700007	100008
	700012	350008
	700015	140008
	700031	24168

Другие варианты заданий № 25

- Другое количество делителей, например, 6

Pascal: if divCount(k) = 6 then, if (divs.Count = 6) then

Python: if divCount(k) == 6:, if (len(divs) == 6):

C++: if (divCount(k) == 6), if (divs.size() == 6)

- Указанное количество чётных / нечётных делителей:
 - При решении через функцию нахождения количества при нахождении количества делителей проверять делители на чётность / нечётность
 - При решении через динамические структуры данных записывать в динамическую структуру (список / вектор) только чётные / нечётные делители
- Задание: измените программы для вывода нечётных делителей чисел отрезка [174457, 174505], имеющих ровно 6 нечётных делителей.
- Подсказка: должна получиться 1 строка при выводе, учитываются все делители.

Выполнение задания на Pascal (функции)

```
function divCount(number: integer): integer;  
var  
    cnt, i: integer;  
begin  
    cnt := 0;  
    for i := 1 to number do  
        if (number mod i = 0) and (i mod 2 = 1) then  
            cnt := cnt + 1;  
    divCount := cnt;  
end;
```

```
procedure divPrint(number: integer);  
var  
    i: integer;  
begin  
    for i := 1 to number do
```

Выполнение задания на Python (структуры)

```
def dividers(number):  
    d = []  
    for i in range(1, number + 1):  
        if number % i == 0 and i % 2 == 1:  
            d.append(i)  
    return d
```

```
for i in range(174457, 174505+1):
```

Другие варианты заданий № 25

- Поиск числа, имеющего максимальное количество делителей

```
def divCount(number):
```

```
    cnt = 0
```

```
    for i in range(1, number + 1):
```

```
        if number % i == 0:
```

```
            cnt += 1
```

```
    return cnt
```

```
def divPrint(number):
```

```
    for i in range(1, number + 1):
```

```
        if number % i == 0:
```

```
            print(i, end = ' ')
```

```
    print()
```

Другие варианты заданий № 25

- Нахождение простых чисел в диапазоне: Python

```
def isPrime(number):  
    prime = True  
    for i in range(2, number):  
        if number % i == 0:  
            prime = False  
            break  
    return prime  
  
for i in range(174457, 174505+1):  
    if isPrime(i):  
        print(i)
```

Другие варианты заданий № 25

- Нахождение простых чисел в диапазоне: C++

```
#include <iostream>
```

```
using namespace std;
```

```
bool isPrime(int number){
```

```
    bool prime = true;
```

```
    for(int i = 2; i < number; i++){
```

```
        if (number % i == 0){
```

```
            prime = false;
```

```
            break;
```

```
        }
```

```
    }
```

```
    return prime;
```

```
}
```

Другие варианты заданий № 25

- Нахождение простых чисел в диапазоне: Pascal

```
function isPrime(number: integer): boolean;
```

```
var
```

```
  i: integer;
```

```
  prime: boolean;
```

```
begin
```

```
  prime := true;
```

```
  for i := 2 to number-1 do
```

```
    if number mod i = 0 then
```

```
      begin
```

```
        prime := false;
```

```
      end;
```

```
  isPrime := prime;
```

```
end;
```

Другие варианты заданий № 25

- Задание: найдите количество пар простых чисел-близнецов в диапазоне $[100000, 101000]$. Выведите числа каждой пары и количество пар.
- Числа-близнецы – пары простых чисел, отличающихся на 2.
- Реализация алгоритма 1: записать в динамическую структуру данных все простые числа в заданном диапазоне, подсчитать количество пар, для которых разность чисел равна 2.
- Реализация алгоритма 2: проверять на простоту числа i и $i+2$, если оба числа простые, изменять счётчик количества пар.

Числа-близнецы (алгоритм 1)

```
def isPrime(number):  
    prime = True  
    for i in range(2, number):  
        if number % i == 0:  
            prime = False  
            break  
    return prime
```

```
a = []
```

Числа-близнецы (алгоритм 2)

```
def isPrime(number):  
    prime = True  
    for i in range(2, number):  
        if number % i == 0:  
            prime = False  
            break  
    return prime
```

```
cnt = 0
```

Другие варианты заданий № 25

- Поиск наибольшего общего делителя, наименьшего общего кратного
- **Наибольший общий делитель** чисел a и b – наибольшее число, которое делит a и b без остатка

```
while(a != b):  
    if a > b:  
        a = a - b  
    else:  
        b = b - a  
print(a)
```

```
while a != 0 and b != 0:  
    if a > b:  
        a = a % b  
    else:  
        b = b % a  
print(a + b)
```

- **Наименьшее общее кратное** чисел a и b – наименьшее число, которое делится на a и b без остатка

```
nok = min(a, b)
```

```
while(True):
```

```
    if nok % a == 0 and nok % b == 0:
```

```
        break
```

```
    nok += 1
```

```
print(nok)
```

Другие варианты заданий № 25

- Задание: выведите числа в диапазоне $[100, 1000]$, имеющие в качестве наибольшего общего делителя с числом 310 число 31.

- Ответ:

217

279

341

403

527

589

651

713

837

899

961

Другие варианты заданий № 25

- Перевод в другие системы счисления

Пример: подсчёт количества нулей и единиц в двоичной записи числа

number = 22 # Здесь может быть ввод данных

```
cnt0 = 0
```

```
cnt1 = 0
```

```
while number > 0:
```

```
    digit = number % 2
```

```
    if digit == 0:
```

```
        cnt0 += 1
```

```
    if digit == 1:
```

```
        cnt1 += 1
```

```
    number = number // 2
```

```
print("Количество 0 =", cnt0)
```

```
print("Количество 1 =", cnt1)
```

Другие варианты заданий № 25

- Задание: выведите числа в диапазоне [10100, 10150], имеющие равное количество нулей и единиц в двоичной записи числа.

- Ответ:

10115

10117

10118

10121

10122

10124

10129

10130

10132

10136

10145

10146

10148