



# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Общински кръг, 7 януари 2024 г.

Група С, 7 – 8 клас

## Задача С2. ПРОСТИ ЧИСЛА (Пояснение към решението)

1. Наивният подход за решаване на задачата, при който първо пресмятаме произведението  $P$  и след това намираме неговите прости множители, ще получи 20 т.

2. По-добър подход, осигуряващ 60 т., е да намираме простите множители последователно на поредните числа, образувачи произведението  $P$ , като отбелязваме само тези, които не се повтарят. Това може да се направи например с използване на типа `set` от STL, реализирано във функцията `div`:

```
set<int>s;
void div(int a)
{ while(a%2==0) {s.insert(2); a = a/2;}
  int p=3;
  while(a>1)
    {while(a%p==0) {s.insert(p); a = a/p;} p += 2;}
}
```

Тази функция извикваме за всеки елемент на произведението  $P$  и накрая отпечатваме размера на множеството `s`:

```
for(int a=a1;a<=a2;a++) div(a);
cout << s.size() << endl;
```

3. Подходът, който осигурява 80 т., е да използваме метода на решето на Ератостен и в масива `s[i]` да заредим `s[i]=0`, когато `i` е просто число и `s[i]=1` в обратния случай. След това чрез следната функция увеличаваме глобалния брояч `c` с единица, когато намерим нов прост делител:

```
void d(int a)
{if(s[a]==0){s[a]=1;c++;}
 else for(int k=2;k*k<=a;k++)
   if(a%k==0)
     {if(s[k]==0) {s[k]=1;c++;}
      if(s[a/k]==0) {s[a/k]=1; c++;}
     }
}
```

Накрая трябва да извикаме в цикъл тази функция:

```
for(int a=a1;a<=a2;a++) d(a);
cout << c << endl;
```

4. За да изкараме 100 т. трябва отново да използваме решето на Ератостен, но накрая да проверим за всяко просто число дали е делител на някое число в интервала  $[a_1; a_2]$ . Това може да стане лесно, като сметнем колко числа се делят на дадено число  $p$  в интервала със стандартната формула -  $\left[ \frac{a_2}{p} - \frac{a_1-1}{p} \right]$ .

Емил Келеведжиев