

## Задача СУПЕР ПРОСТИ ЧИСЛА

### Пояснение към решението

#### Бавно решение

За всяка заявка пробягваме числата от  $L$  до  $R$  и проверяваме, кои от тези числа са прости. За целта използваме функцията:

```
bool isPrime(int a)
{
    for(int i=2;i*i<=a;i++)
        if(a%i==0) return false;
    return true;
}
```

В брояча  $c$  се получава броят на простите числа от числовия интервал. След това проверяваме дали този брой е просто число и ако е така, увеличаваме друг брояч  $r$ :

```
if(isPrime(c)) r++;
```

Накрая отпечатваме  $r$ .

#### По-бързо решение

Използваме идеята на бавното решение, но вместо всеки път, когато трябва да проверим, дали едно число  $j$  е просто, не извикваме функция, а проверяваме предварително маркирани стойности на масива  $s[j]$ . Масива го запълваме с `true` или `false` според това, дали  $j$  е просто или не е. За целта използваме метода на решетото на Ератостен, където  $m$  има максималната стойност на числата, които ще проверяваме:

```
void sieve()
{
    fill(s+2, s+m, 1);
    for (int i = 2; i*i <= m; i++)
        if (s[i] == 1)
            for (int j = 2*i; j <= m; j += i) s[j] = 0;
}
```

**Бързо решение** За да намалим времето за работа на програмата, използваме спомагателен масив  $p[i]$ , в който за всяко  $i$  предварително еднократно пресмятаме колко супер прости числа има в интервала от 1 до  $i$ . Това става чрез функцията `count`:

```
void count()
{
    int c = 0;
    for (int i = 0; i <= m; i++)
    {
        if (s[i]) c++;
        if (s[c]) p[i] = 1;
    }
    for (int i = 1; i <= m; i++) p[i] += p[i-1];
}
```

Тогава броят на супер простите числа в интервала от  $L$  до  $R$  е равен на

$$p[R] - p[L-1]$$

*Зорница Дженова*