

Задача С?. ОПАШКА ОТ НУЛИ

Пояснение към решението

Бавно решение – файл `tail_41.cpp`

Програмата прочита входните данни в променливите `a` и `b`. Променливите `c1`, `c2` и `c3` служат за пресмятане на търсените бройки, които трябва да се изведат съответно на първия, втория и третия ред в изхода.

Използваме цикъл `for`, в който променливата `i` пробягва последователно стойностите от `a` до `b`. За всяка такава стойност `i` в тялото на цикъла се извършва следното:

Когато остатъкът от делението на `i` с `10` не е нула, това означава че числото `i` не завършва на `0` и затова преминаваме към следващата стойност на `i`:

```
if(i%10!=0) continue;
```

Когато остатъкът от делението на `i` с `10000` е нула, това означава че числото `i` завършва на повече от три нули и затова преминаваме към следващата стойност на `i`:

```
if(i%10000==0) continue;
```

В останалите случаи използваме следния оператор:

```
if(i%1000==0) c3++;  
else if(i%100==0) c2++;  
else c1++;
```

Той извършва следното:

За стойността на `i` проверяваме първо дали `i` се дели на `1000`, което означава, че `i` завършва на точно три нули и тогава увеличаваме брояча `c3`.

За стойностите на `i`, които не завършват на три нули, проверяваме дали завършват на 2 нули и тогава увеличаваме брояча `c2`.

За останалите стойности (те завършват на точно една нула) увеличаваме брояча `c1`.

Броят на операциите в това решение е пропорционален на стойността на разликата `b-a`.

Бързо решение – файл `tail.cpp`

Функцията `f(b, c[])` пресмята в елементите на масива `c[i]` колко са целите числа от `1` до `b`, включително, които завършват точно на `i` нули.

В тялото на функцията първо се пресмята `d`, равно на 10^{18} и след това стойностите `c[i]` се зарежда с броя на числата от интервала от `1` до `b`, които се делят без остатък на 10^i за $i = 10^{18}, 10^{17}, \dots, 10^1$. След това, съответният брой на числата от интервала от `1` до `b`, които завършват точно на `i` нули се пресмята в `c[i]` чрез следния фрагмент, където `NZ` е `18` и `s` започва със стойност `0`:

```
for(int i=NZ; i>0; i--)  
{  
    s=s+c[i+1];  
    c[i]=c[i]-s;  
}
```

В главната програма пресмятаме $f(a-1, c1); f(b, c2);$ където $c1$ и $c2$ са масиви и отпечатваме

```
for(int i=1;i<=3;i++) cout << c2[i]-c1[i] << endl;
```

Броят на операциите в това решение е пропорционален на максималния брой ($NZ=18$) на цифрите в b .

Емил Келеведжиев