

Задача ПРЕБРОЯВАНЕ

Пояснение към решението

Бавно решение с изчерпващо търсене

Пресмятаме границите на интервала от целите числа, които са n -цифрени в бройна система с основа p :

```
long long int d=1;
for(int i=1;i<n;i++) d=p*d;
```

Така границите на този интервал са стойностите на d и $p*d - 1$. За всяко число от този интервал намираме сбора от цифрите му и сравняваме дали този сбор е равен на s :

```
long long int c=0;
for(long long int i=d; i<p*d; i++)
{
    long long int ss=0;
    long long int a=i;
    while(a>0)
        { ss += a%p; a=a/p;}
    if(ss==s) c++;
}
```

Накрая отпечатваме стойността на брояча c .

По-бързо решение с рекурсия

Използвана е функцията $run(n, s)$, която връща броя на разглежданите числа, които имат n цифри и сумата от цифрите им е равна на s . Функцията е програмирана рекурсивно, като при $n==1$ връща 1, когато $s < p$, а в противен случай връща 0. Това е очевидно за едноцифрените числа. При $n > 1$ функцията връща стойността на c , която се пресмята от

```
LL c=0;
for(int i=1;i<p;i++)
    if(i <= s) c += run(n-1,s-i);
```

Тук е използван типът LL , дефиниран чрез `typedef long long int LL;` В главната функция `main()`, понеже старшата цифра не може да бъде нула, функцията `run` се извиква в следния цикъл:

```
LL c=0;
for(int i=1;i<p;i++)
    if(i <= s) c += run(n-1,s-i);
```

и се отпечатва стойността на c.

Бързо решение чрез рекурсия с мемоизация (рекурсия с таблица)

Използва се подход с въвеждане на спомагателен масив със стойности `t[n][s]`, в които се запомня стойността, която ще върне извикването на `run(n, s)`. Така първоначално стойностите на спомагателния масив се зареждат с числото `-1` чрез `memset(t, -1, sizeof(t));` и в тялото на функцията `run` рекурсивното развитие се отсича чрез оператора

```
if (t[n][s] != -1) return t[n][s];
```

Емил Келеведжиев