

Задача СУМИ

Пояснение към решенията

Бавно решение (sums_46p.cpp)

Елементите на дадената редица се прочитат в масива $a[i]$. Дадените двойки индекси чрез цикъла `for(int j=1; j<=k; j++)` се прочитат съответно в масивите $L[j]$ и $R[j]$ за определяне на левия и десния край на разглежданите подредици. Функцията

```
int sumdig(int a)
{ int s=0;
  while(a>0)
  {
    s+=a%10;
    a=a/10;
  }
  return s;
}
```

пресмята сумата от цифрите на положителното число a .

Функцията `solve_sum()` пресмята и отпечатва сумата s от всички суми на подредиците, определени чрез двойките индекси $L[j]$ и $R[j]$. Това се извършва чрез цикъл с индексна променлива k и вложен цикъл с индексна променлива i , с който се пробягват елементите на подредиците.

Функцията `solve_sum_digits()` работи аналогично на `solve_sum()`, но вместо да натрупва сумите от елементите $a[i]$ на подредиците, натрупва сумите от сумите на цифрите на числата $a[i]$:

```
s = s + sumdig(a[i]);
```

Програмата работи бавно, защото в нея има тройно вложен цикъл.

Решение за 100 т. (sums_100p.cpp)

За да се реализира програма с двойно вложен цикъл, първо в масивите $ps[i]$ и $pd[i]$ се пресмятат съответно сумите на първите i елемента на дадената редица и на сумите от сумите от цифрите на първите i елемента. Така сумата на елементите на позиции от i_1 до i_2 се пресмята с израза $ps[i_2]-ps[i_1]$, а сумата от сумите на цифрите на елементите на позиции от i_1 до i_2 – с израза $pd[i_2]-pd[i_1]$.

В цикъла `for(int j=1; j<=k; j++)` се прочитат последователно двойките индекси i_1 и i_2 , определящи краищата на подредиците. Във вложения цикъл се пресмятат двете суми, които се извеждат на първия и на втория ред в изхода:

```
for(int i=i1; i<=i2; i++)
{
  ss = ss + ps[i]-ps[i-1];
  sd = sd + pd[i]-pd[i-1];
}
```

Накрая се отпечатват натрупаните стойности на ss и sd .

Емил Келеведжиев