

## Задача D1. ПРЕМАХВАНЕ

### Пояснение към решенията

#### Бавно решение $O(n^3)$ – файл `removal_n3.cpp`

Програмата прочита в масива `a[]` елементите на дадената редица. Пресмята сумата им в променливата `t`. Ако сумата е равна на `k`, очевидно отговорът на задачата е стойността на `n`.

Основната част на програмата е двоен цикъл в който индексът `i` описва начален отрез от масива `a[]`, а индексът `j` – краен отрез.

```
for(int i=0;i<n;i++)  
for(int j=n-1;j>=i;j--)
```

С този цикъл правим пълно изчерпване на всички възможности за премахване на елементи от дадената редица. В тялото му сумиране в `s` елементите на масива, заключени между индексите `i` и `j`. За целта използваме още един цикъл:

```
int s=0;  
for(int p=i;p<=j;p++) s=s+a[p];
```

Понеже `t-s` дава сумата на премахнатите елементи, пресмятаме в променливата `r` най-малкия им брой, такъв че сумата на премахнатите да е равна на `k`:

```
if(k==t-s) r=min(r,i+(n-j-1));
```

Накрая отпечатваме

```
if(r==n+1) cout << -1 << endl;  
else cout << r << endl;
```

#### Оптимизация на бавното решение $O(n^2)$ – файл `removal_n2.cpp`

За да увеличим бързодействието, премахваме цикъла, с който пресмятаме сумата от елементите на масива, заключени между индексите `i` и `j`. За целта може да използваме масив `s[i]`, в който поддържа сумата на елементите с индекси от `i` до `n-1`, а в променливата `ss` – сумата на елементите с индекси от `i` до `j`. Това става чрез изваждане на елемент от масива в тялото на цикъла по `i`:

```
s[i+1]=s[i]-a[i];
```

и в тялото на цикъла по `j`:

```
ss -= a[j];
```

#### Бързо решение $O(n)$ – файл `removal.cpp`

В началото на програмата, също както в бавното решение, пресмятаме в `t` сумата от елементите на масива `a[]`.

Забелязваме, че най-малък брой премахнати елементи със сума `k` ще имаме, когато намерим най-голям брой непремахнати елементи, чиято сума е `t-k`.

Затова започваме да движим по масива прозорец с ляв край, определен от индекса `i` и с десен край – определен от индекса `j`. Двата индекса ги движим само надясно. В променливата `cs` се поддържа сумата от елементите, намиращи се в прозореца, а в променливата `m` се пресмята най-големия брой елементи, попадащи в прозореца така, че `cs` да е равно на `s`, където `s=t-k`.

```
for (int j = 0; j < n; j++)
{
    cs += a[j];
    while (i < j && cs > s) {cs -= a[i]; i++;}
    if (cs == s) m = max(m, j - i + 1);
}
```

Накрая отпечатваме

```
if(m==-1) cout << -1; else cout << n-m;
cout << endl;
```

*Зорница Дженкова*