Olimpiadi Italiane di Informatica – Salerno 2013

# Spiedini di frutta (`spiedini`)

## Descrizione del problema

Ad Elisa piace molto la frutta, ma è anche molto esigente: il suo frutto preferito è la fragola, di cui non si stancherebbe mai; al contrario, riesce a mangiare solo un certo numero di altri frutti.

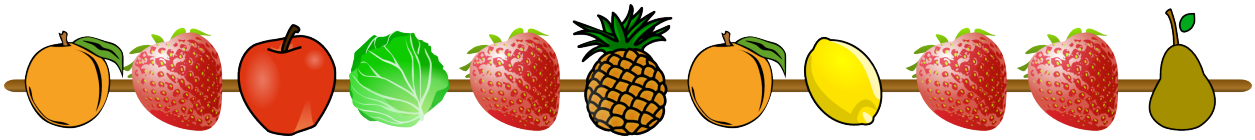
In particolare, a ogni frutto Elisa assegna un valore che indica quanto quel frutto non le piace: 0 per le fragole, altrimenti un numero intero compreso tra 1 (sopportabile) e 9 (lo detesta). In un pasto, Elisa può mangiare dei frutti solo se la somma dei loro valori non supera $K$.

A una cena, Elisa ordina uno spiedino di $N$ frutti; sapendo che è possibile estrarre i frutti solo dagli estremi, e che Elisa deve mangiare ogni frutto che estrae dallo spiedino, aiutate Elisa a scegliere la strategia migliore per mangiare quante più fragole possibile.

Per esempio, consideriamo il seguente spiedino con 11 frutti, dove la prima riga indica le posizioni dei frutti nello spiedino, numerate da 0 a 10, mentre la seconda riga contiene il valore del frutto (ricorda che il valore 0 corrisponde sempre a una fragola):

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 2 | 8 | 0 | 5 | 1 | 6 | 0 | 0 | 3 |

Questa tabella descrive, per esempio, il seguente spiedino:



Supponiamo che $K$ sia uguale a 9. La strategia migliore per Elisa è quella di mangiare i frutti fino alla posizione 1 da sinistra, e fino alla posizione 8 da destra. In questa maniera mangia 3 fragole (nelle posizioni 1, 8 e 9), mentre la somma dei valori dei frutti che non le piacciono è 4 (1 in posizione 0 e 3 in posizione 10). Per riuscire a mangiare anche l'ultima fragola sarebbe necessario $K \geq 14$.

Per risolvere il problema dovete scrivere una funzione `solve`$(N, K, S)$ che restituisca un singolo numero intero: il massimo numero di fragole che Elisa può mangiare, sapendo che lo spiedino è composto da $N$ frutti aventi valori $S_0, S_1, \ldots, S_{N-1}$, e sapendo che Elisa tollera al massimo un valore di $K$.

## Assunzioni

Per tutti i subtask vale l'assunzione $0 \leq K \leq 1\,000\,000\,000$.

## Subtask

- **Subtask 0 [5 punti]:** caso di esempio.
- **Subtask 1 [10 punti]:** $1 \leq N \leq 500$.
- **Subtask 2 [27 punti]:** $1 \leq N \leq 5\,000$.
- **Subtask 3 [48 punti]:** $1 \leq N \leq 1\,000\,000$.
- **Subtask 4 [10 punti]:** $1 \leq N \leq 20\,000\,000$.

## Dettagli di implementazione

Dovrai sottoporre esattamente un file con estensione: `.c`, `.cpp` o `.pas`. Questo file deve implementare la funzione `solve` utilizzando uno dei seguenti prototipi.

### Programma in C o C++

```
int solve(int N, int K, int* S);
```

### Programma in Pascal

```
function solve(N: longint; K: longint; var S: array of longint): longint;
```

La funzione implementata dovrà comportarsi come sopra descritto. Naturalmente sei libero di implementare altre routine ausiliarie per uso interno. La tua sottoposizione non deve effettuare direttamente operazioni di input o output, via console o via file di testo.

### Funzionamento del grader di esempio

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati di input dal file `input.txt`, chiama la funzione che dovete implementare, e scrive il risultato restituito dalla vostra funzione sul file `output.txt`. Il file `input.txt` deve essere composto da 2 righe, in questo formato:

- nella prima riga devono essere presenti 2 interi: $N$ e $K$;
- nella seconda riga devono essere presenti $N$ interi: i valori dei frutti: $S_0$, $S_1$, ..., $S_{N-1}$.

## Esempio di input/output

| **File** `input.txt` | **File** `output.txt` |
|---|---|
| 11 9<br>1 0 2 8 0 5 1 6 0 0 3 | 3 |