

## Finanza creativa (bilancio)

Limite di tempo: 1.0 secondi  
Limite di memoria: 256 MiB

La *SteamPower S.P.A.*, azienda leader mondiale nel campo delle macchine a vapore portatili, non accenna ad uscire dal periodo di crisi nonostante i forti e ben ponderati tagli al personale di recente effettuati. Per fortuna il *CEO* ha avuto una nuova geniale idea: affidarsi al massimo esperto mondiale in campo di finanza creativa. L'esperto ha già ricevuto il bilancio dal reparto contabilità, e la situazione è disastrosa: arrivati a questo punto l'unico modo che conosce per rassicurare gli azionisti e prendere tempo è quello di effettuare qualche piccolo ritocco ai libri contabili.

Più precisamente, vuole ricorrere al suo fedele bianchetto per correggere il totale  $U$  delle uscite. Inoltre, grazie agli insegnamenti di lunghi anni di esperienza, sa che per contenere i rischi dell'operazione non deve assolutamente eccedere le  $K$  cifre cancellate (sulle  $N$  complessive del totale  $U$ ).

Aiuta l'esperto di finanza creativa a trovare il minimo intero ottenibile cancellando  $K$  cifre dall'intero  $U$ !

## Implementazione

Dovrai sottoporre esattamente un file con estensione `.c`, `.cpp` o `.pas`.

📎 Tra gli allegati a questo task troverai un template (`bilancio.c`, `bilancio.cpp`, `bilancio.pas`) con un esempio di implementazione da completare.

Se sceglierai di utilizzare il template, dovrai implementare la seguente funzione:

C/C++	<code>void bianchetta(int N, int K, int U[], int C[]);</code>
Pascal	<code>procedure bianchetta(N, K: longint; var U, C: array of longint);</code>

In cui:

- L'intero  $N$  rappresenta il numero di cifre del totale delle uscite.
- L'intero  $K$  rappresenta il numero di cifre da cancellare.
- L'array  $U$ , indicizzato da 0 a  $N - 1$ , contiene l'elenco delle cifre del totale delle uscite, con la cifra più significativa in posizione 0 e così via fino alla cifra delle unità in posizione  $N - 1$ .
- La funzione dovrà riempire l'array  $C$ , indicizzato da 0 a  $N - K - 1$ , con l'elenco delle cifre rimaste dopo la cancellazione (dalla più alla meno significativa), che verrà stampato sul file di output.

## Dati di input

Il file `input.txt` è composto da due righe. La prima riga contiene i due interi  $N$  e  $K$ . La seconda riga contiene le  $N$  cifre  $U_i$  del totale delle uscite  $U$ , separate da uno spazio.

## Dati di output

Il file `output.txt` è composto da un'unica riga contenente  $N - K$  cifre separate da uno spazio, la risposta a questo problema.

## Assunzioni

- $1 \leq K < N \leq 1\,000\,000$ .
- $0 \leq U_i \leq 9$  per ogni  $i = 0 \dots N - 1$ , e  $U_0 \geq 1$ .

## Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [20 punti]:**  $N \leq 10$ .
- **Subtask 3 [30 punti]:**  $N \leq 200$ .
- **Subtask 3 [20 punti]:**  $N \leq 10\,000$ .
- **Subtask 4 [20 punti]:** Nessuna limitazione specifica.

## Esempi di input/output

input.txt	output.txt
5 2 1 9 5 0 3	1 0 3
input.txt	output.txt
9 3 9 4 1 5 7 1 1 2 3	1 5 1 1 2 3
input.txt	output.txt
10 5 1 3 2 0 2 1 8 5 3 6	0 1 5 3 6

## Spiegazione

Nel **primo caso di esempio** conviene cancellare le due cifre più alte (5 e 9).

Nel **secondo caso di esempio** conviene cancellare le due cifre più significative (9 e 4), e la restante cifra più alta (il 7).

Nel **terzo caso di esempio** conviene cancellare le tre cifre più significative (1, 3 e 2), il 2 ancora successivo e la restante cifra più alta (l'8).