

## La camera dei cestini (cestini)

Tommaso ha  $N$  oggetti e  $M$  cestini in camera, numerati rispettivamente da 0 a  $N - 1$  e da 0 a  $M - 1$ . Ogni cestino ha una capacità illimitata e vi è un ordine fra gli elementi al suo interno: l'elemento in posizione 0 è sul fondo, quello in posizione 1 si trova subito sopra, e così via.

Inizialmente, tutti gli oggetti si trovano in ordine nel cestino 0, con l'oggetto 0 sul fondo e l'oggetto  $N - 1$  in cima. Tuttavia, quando è annoiato, Tommaso passa il tempo spostando gli oggetti fra i cestini! In particolare, sposta l'oggetto in cima a un cestino in cima ad un altro.

Aiuta Tommaso a tenere traccia degli oggetti nella sua camera, stando sempre pronto a dirgli qual è l'oggetto in una certa posizione di un certo cestino, gestendo i suoi  $Q$  spostamenti e domande.

## Implementazione

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

📖 Tra gli allegati a questo task troverai un template `cestini.c` o `cestini.cpp` con un esempio di implementazione.

Dovrai implementare le seguenti funzioni:

C/C++ | `int inizia(int N, int M);`

- L'intero  $N$  rappresenta il numero di oggetti.
- L'intero  $M$  rappresenta il numero di cestini.

C/C++ | `void sposta(int a, int b);`

- L'intero  $a$  rappresenta l'indice del cestino di partenza,  $b$  quello di arrivo.

C/C++ | `int controlla(int a, int i);`

- La funzione dovrà restituire l'indice dell'elemento del cestino  $a$  in posizione  $i$  dal fondo.
- Se un tale oggetto non esiste (quindi il cestino  $a$  ha meno di  $i + 1$  oggetti), la funzione dovrà restituire  $-1$ .

Il grader chiamerà prima la funzione `inizia`, poi varie volte le funzioni `sposta` e `controlla` e ne stamperà il valore restituito sul file di output.

## Grader di prova

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 da `stdin`, chiama le funzioni che dovete implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da due righe, contenenti:

- Riga 1: gli interi  $N$ ,  $M$  e  $Q$ .
- Righe 2, 3, ...,  $Q + 1$ : la descrizione di uno spostamento o di un controllo, che può quindi essere:
  - `s a b`: una chiamata alla funzione `sposta` con i parametri  $a$  e  $b$ ;
  - `c a i`: una chiamata alla funzione `controlla` con i parametri  $a$  e  $i$ ;

Il file di output è composto da  $C$  righe, dove  $C$  è il numero di chiamate alla funzione `controlla`:

- Riga  $i$ : il valore restituito dall' $i$ -esima chiamata alla funzione `controlla`.

## Assunzioni

- $1 \leq N \leq 100\,000$ .
- $1 \leq M \leq 100\,000$ .
- $1 \leq Q \leq 100\,000$ .
- $0 \leq a \leq M - 1$ ,  $0 \leq i \leq N - 1$  per ogni chiamata alla funzione `controlla`.
- $0 \leq a, b \leq M - 1$ ,  $a \neq b$  per ogni chiamata alla funzione `sposta`.
- È garantito che ad ogni chiamata alla funzione `sposta`, il cestino  $a$  ha almeno un oggetto.

## 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 che lo compongono.

- **Subtask 1** [ 0 punti]: Casi d'esempio.
- **Subtask 2** [20 punti]:  $M \leq 2$ .
- **Subtask 3** [10 punti]:  $N \leq 10$ .
- **Subtask 4** [30 punti]:  $Q \leq 100$ .
- **Subtask 5** [40 punti]: Nessuna limitazione aggiuntiva.

## Esempi di input/output

stdin	stdout
3 3 2 s 0 1 c 1 0	2
5 6 7 s 0 1 c 1 0 s 0 2 s 1 2 s 0 2 c 2 2 c 2 1	4 2 4

# Spiegazione

Nel **primo caso di esempio** l'oggetto 2, in cima al cestino 0, viene spostato in cima al cestino 1.

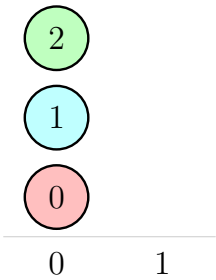


Figura 1: la disposizione iniziale

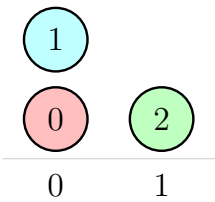


Figura 2: la disposizione dopo il primo spostamento

Nel **secondo caso di esempio**:

- l'oggetto 4 viene spostato dal cestino 0 al cestino 1;
- l'oggetto 3 viene spostato dal cestino 0 al cestino 2;
- l'oggetto 4 viene spostato dal cestino 1 al cestino 2;
- l'oggetto 2 viene spostato dal cestino 0 al cestino 1.

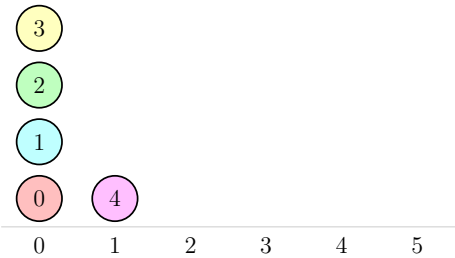


Figura 3: la disposizione dopo il primo spostamento

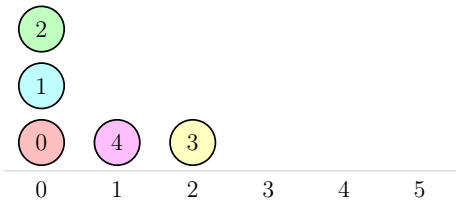


Figura 4: la disposizione dopo il secondo spostamento

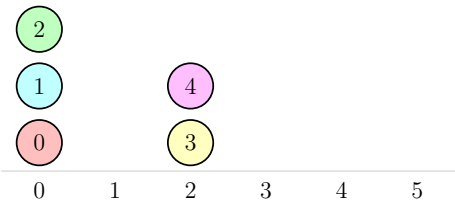


Figura 5: la disposizione dopo il terzo spostamento

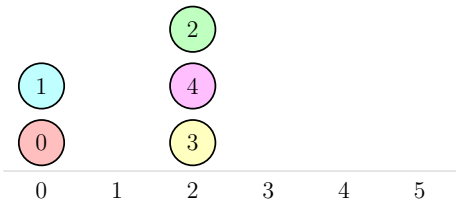


Figura 6: la disposizione dopo il quarto spostamento