

Scatole di souvenir

L'ultimo numero della cerimonia di apertura delle IOI 2015 è in corso, e ora ciascun team deve ricevere una scatola di souvenir da parte dell'host. Tuttavia, le guide sono così affascinate dalla cerimonia che si sono dimenticate dei souvenir! L'unica persona che se ne ricorda ancora è Aman, una esuberante guida che vuole che le IOI si svolgano in maniera impeccabile. Aman ha quindi intenzione di consegnare personalmente tutti i souvenir nel minor tempo possibile.

La sede dove si svolge la cerimonia è una circonferenza suddivisa in L settori identici, numerati da 0 a $L - 1$. Vale a dire: per $0 \leq i \leq L - 2$, il settore i è adiacente al settore $i + 1$, inoltre il settore $L - 1$ è adiacente al settore 0 . Sono presenti N team alla cerimonia e ciascuno è situato in uno dei settori, i quali possono contenere un numero arbitrario di team. Alcuni settori potrebbero essere vuoti. Ci sono N souvenir identici, che all'inizio si trovano con Aman nel settore 0 . Aman deve consegnare un souvenir a ciascun team, e ritornare poi al settore 0 .

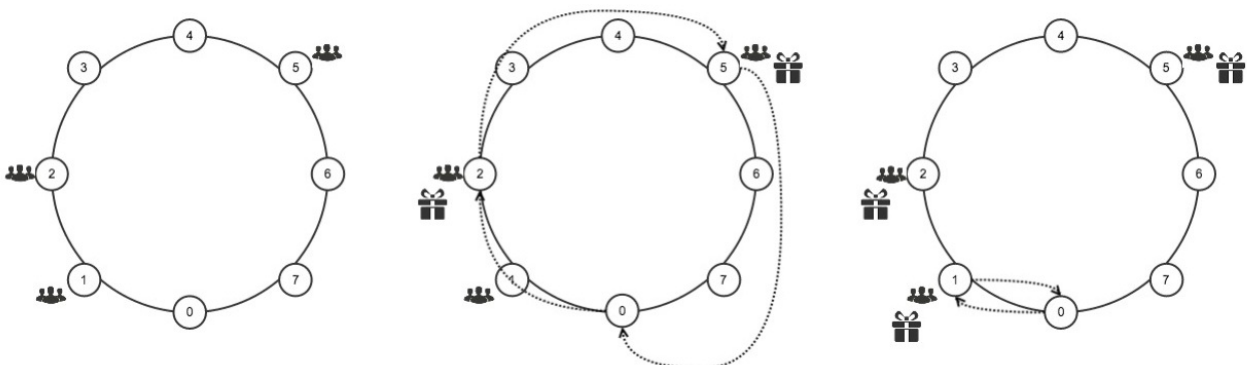
Nota: alcuni team potrebbero essere situati nel settore 0 .

In un dato istante, Aman può trasportare al massimo K souvenir. I souvenir si possono prendere solo nel settore 0 , e fare ciò non richiede tempo. Ogni souvenir deve essere trasportato fino alla sua consegna. In qualunque momento Aman si trovi in un settore in cui è presente un team che non ha ancora ricevuto il souvenir, può decidere di lasciargliene uno (se ce l'ha). Anche questa operazione non richiede tempo, e può essere ripetuta per ciascun team incontrato. L'unica operazione che richiede tempo è lo spostamento tra i settori.

Aman può muoversi lungo la circonferenza dei settori in entrambe le direzioni. Per spostarsi (in entrambi i versi) verso un settore adiacente è necessario esattamente un secondo, indipendentemente dal numero di souvenir trasportati. Il tuo obiettivo è trovare il minimo numero di secondi necessari affinché Aman riesca a consegnare tutti i souvenir e a ritornare alla sua posizione iniziale.

Esempio

In questo esempio abbiamo $N = 3$ team, la capacità di trasporto di Aman è pari a $K = 2$, ed il numero di settori è $L = 8$. I team sono situati nei settori 1 , 2 , e 5 .



Una delle soluzioni ottimali è quella raffigurata sopra: nel suo primo viaggio, Aman porta con sé due souvenir, ne consegna uno all'unico team che si trova nel settore **2**, poi un altro al team del settore **5**, per poi tornare alla sezione **0**. Questo viaggio richiede **8** secondi. Nel suo secondo viaggio, Aman porta l'ultimo souvenir al team del settore **1** e poi ritorna al settore **0**, stavolta impiegando **2** secondi. Il totale, quindi, è di **10** secondi.

Implementazione

Dati i tre valori N , K , L , e le posizioni di tutti i team, devi calcolare il minimo numero di secondi necessari ad Aman per consegnare tutti i souvenir e ritornare al settore **0**. Dovrai implementare la funzione `delivery`:

- `delivery(N, K, L, positions)` — Questa funzione verrà invocata dal grader esattamente una volta.
 - N : il numero di team.
 - K : il massimo numero di souvenir che Aman riesce a trasportare.
 - L : il numero di settori nella sede della cerimonia di apertura.
 - `positions`: un array di lunghezza N , in cui `positions[0]`, ..., `positions[N-1]` forniscono i numeri di settore di tutti i team. Gli elementi di `positions` sono in ordine non decrescente.
 - La funzione deve restituire il minimo numero di secondi sufficienti ad Aman per completare la consegna dei souvenir.

Subtask

subtask	punti	N	K	L
1	10	$1 \leq N \leq 1000$	$K = 1$	$1 \leq L \leq 10^9$
2	10	$1 \leq N \leq 1000$	$K = N$	$1 \leq L \leq 10^9$
3	15	$1 \leq N \leq 10$	$1 \leq K \leq N$	$1 \leq L \leq 10^9$
4	15	$1 \leq N \leq 1000$	$1 \leq K \leq N$	$1 \leq L \leq 10^9$
5	20	$1 \leq N \leq 10^6$	$1 \leq K \leq 3000$	$1 \leq L \leq 10^9$
6	30	$1 \leq N \leq 10^7$	$1 \leq K \leq N$	$1 \leq L \leq 10^9$

Grader di prova

Il grader di prova legge l'input nel seguente formato:

- riga 1: $N K L$
- riga 2: `positions[0] ... positions[N-1]`

Il grader di prova scrive in output il valore di ritorno di `delivery`.