

## Il parcheggio

### Il problema

Un parcheggio vicino alla Grande Muraglia ha una lunga fila di posti macchina. Un estremo della fila è quello sinistro, e l'altro è quello destro. La fila è piena di macchine. Ogni macchina ha un tipo e più macchine possono essere dello stesso tipo. I tipi sono identificati da numeri interi. Un gruppo di parcheggiatori decide di riordinare le macchine parcheggiate nella fila in ordine crescente (da sinistra a destra) di tipo di macchina, usando il metodo seguente. In quello che chiameremo un *turno*, ogni parcheggiatore può far uscire una macchina dal suo posto e parcheggiarla in un posto da cui una macchina è stata fatta uscire nello stesso turno. Può succedere che, durante qualche turno, qualcuno dei parcheggiatori non sposti alcuna macchina. Per efficienza, vogliamo utilizzare un numero di turni basso.

Supponiamo che  $N$  sia il numero di macchine e  $W$  il numero di parcheggiatori. Dovete scrivere un programma che, dati i tipi delle macchine parcheggiate e il numero di parcheggiatori, trovi un modo di ordinare le macchine che impieghi un numero di turni minore o uguale a  $\lceil N/(W-1) \rceil$ , vale a dire  $N/(W-1)$  arrotondato per eccesso. Il minimo numero di turni non è mai superiore a  $\lceil N/(W-1) \rceil$ .

Considerate il seguente esempio. Ci sono 10 macchine parcheggiate, di tipo 1, 2, 3 e 4, e ci sono 4 parcheggiatori. Il posizionamento iniziale delle macchine (identificate ciascuna dal proprio tipo), da sinistra a destra, è

2 3 3 4 4 2 1 1 3 1

Il minimo numero di turni è 3, e i turni possono essere realizzati in modo che il posizionamento dopo ogni turno sia come segue:

2 1 1 4 4 2 3 3 3 1 dopo il primo turno;  
 2 1 1 2 4 3 3 3 4 1 dopo il secondo turno;  
 1 1 1 2 2 3 3 3 4 4 dopo il terzo turno.

### Dati in input

Il file di input si chiama `CAR.IN`. La prima riga del file contiene tre numeri interi. Il primo intero è il numero di macchine  $N$ ,  $1 \leq N \leq 20000$ . Il secondo intero è il numero di tipi  $M$ ,  $2 \leq M \leq 50$ . I tipi di macchine sono identificati con gli interi da 1 a  $M$ . C'è almeno una macchina di ogni tipo. Il terzo intero è il numero di parcheggiatori  $W$ ,  $2 \leq W \leq M$ . La seconda riga contiene  $N$  interi, dove l' $i$ -esimo intero è il tipo della  $i$ -esima macchina nella fila, partendo dall'estremo sinistro.

## Dati in output

Il file di output si chiama `CAR.OUT`. La prima riga contiene un numero intero  $R$ , che è il numero di turni della soluzione. Le seguenti  $R$  righe descrivono i turni, ordinati da 1 a  $R$ . In ogni riga il primo intero è il numero di macchine  $C$  che vengono spostate in quel turno; a questo intero seguono  $2C$  interi, che identificano le posizioni delle macchine (le posizioni delle macchine sono identificate da interi tra 1 e  $N$ , a partire dall'estremo sinistro). I primi due interi sono una coppia che descrive come una delle auto viene spostata: il primo intero della coppia è la posizione dall'estremo sinistro *prima* del turno, e il secondo è la posizione dall'estremo sinistro *dopo* il turno. I due interi che seguono formano una coppia che descrive come un'altra macchina viene spostata, e così via. Ci possono essere molte soluzioni diverse per queste  $R$  righe, e il vostro programma deve emetterne in output solo una.

## Esempio di input e output

CAR . IN	CAR . OUT
10 4 4	3
2 3 3 4 4 2 1 1 3 1	4 2 7 3 8 7 2 8 3
	3 4 9 9 6 6 4
	3 1 5 5 10 10 1

## Punteggio parziale

Supponiamo che l'output del vostro programma su un certo input sia  $R$  e che  $\lceil N/(W - 1) \rceil$  sia  $Q$ . Se nell'output del vostro programma gli  $R$  turni non sono descritti correttamente o non producono l'ordine delle macchine richiesto, il vostro punteggio è 0. Altrimenti, il vostro punteggio verrà calcolato sulla base del punteggio massimo come segue:

- $R \leq Q$ : 100% del punteggio;
- $R = Q + 1$ : 50% del punteggio;
- $R = Q + 2$ : 20% del punteggio;
- $R \geq Q + 3$ : 0% del punteggio.