Chess Rush

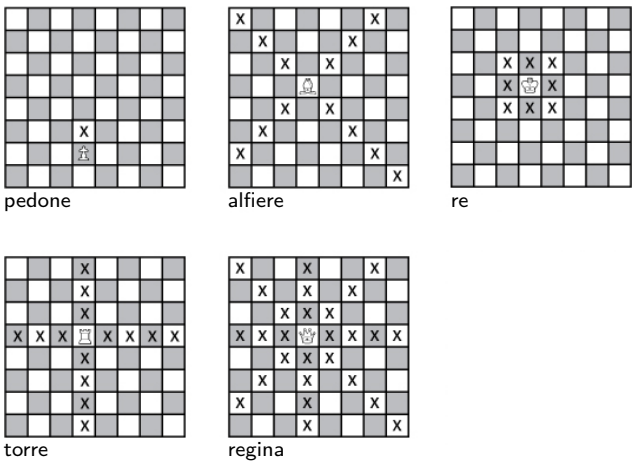
Il mitico mondo di Chess Land è una griglia rettangolare, formata da caselle quadrate, con R righe e C colonne, dove R è maggiore o uguale a C. Le righe e le colonne sono rispettivamente numerate da 1 a R e da 1 a C.

Gli abitanti di Chess Land sono solitamente chiamati *pezzi* nel linguaggio comune; ci sono 5 tipi specifici di abitanti che vagano per il paese: pedoni, torri, alfieri, regine e re. Contrariamente a quanto si possa credere, la cavalleria è morta da tempo a Chess Land, quindi non ci sono cavalli.

Ogni pezzo si muove in modo diverso da una casella a un'altra: in una mossa,

- un pedone si può muovere di una riga in avanti (ovvero dalla riga r alla riga $r + 1$), senza cambiare colonna;
- una torre si può muovere di un qualunque numero di colonne a destra o a sinistra, senza cambiare riga, oppure di un qualunque numero di righe in avanti o indietro, senza cambiare colonna;
- un alfiere può muoversi in una qualunque casella delle due due diagonali passanti per la casella da lui occupata;
- una regina si può muovere in una qualunque casella in cui una torre o un alfiere si possono muovere se si trovassero nella posizione della regina;
- un re si può muovere in una delle 8 caselle a lui adiacenti.

Nella figura seguente, sono segnate con una X le caselle che ogni pezzo può raggiungere in una singola mossa (in questo caso, le righe sono numerate dal basso verso l'alto e le colonne da sinistra a destra).



Di recente, Chess Land è diventato un posto pericoloso: può capitare che pezzi che viaggiano per il paese vengano catturati in maniera inaspettata da forze sconosciute, sparendo nel nulla. Di conseguenza, i pezzi vorrebbero raggiungere le loro destinazioni il più velocemente possibile (ovvero nel numero minore di mosse). Non solo, sono anche interessati al numero di percorsi diversi che raggiungono le destinazioni usando il numero minimo di mosse, dato che la probabilità di essere catturati diminuisce più percorsi ci sono. Due percorsi vengono considerati diversi se differiscono almeno per una casella.

Per questo problema, supponiamo che i pezzi entrino a Chess Land in una data colonna della riga 1, e lascino il paese in una data colonna della riga R. Il tuo compito è rispondere a Q richieste: data la tipologia di un pezzo, la colonna da cui entra nella riga 1 e la colonna che deve raggiungere nella riga R per lasciare il paese, calcola il minimo numero di mosse da compiere e quanti percorsi minimi sono presenti.

Input

La prima riga contiene tre interi R, C e Q, separati da spazio: il numero di righe e di colonne di Chess Land e il numero di richieste, rispettivamente. Seguono Q righe.

Ogni riga è composta da:

- un carattere T, indicante il tipo del pezzo in questione ('P' nel caso di un pedone, 'R' per la torre, 'B' per l'alfiere, 'Q' per la regina e 'K' per il re);
- due interi c_1 e c_R, $1 \leq c_1, c_R \leq C$, che indicano che il pezzo inizia dalla c_1-esima colonna della riga 1 e che deve deve raggiungere la c_R-esima colonna della riga R.

Output

Devi stampare Q righe, l'i-esima riga contiene due interi separati da spazio, ovvero la risposta alla i-esima richiesta: il primo indica il numero minimo di mosse necessarie, il secondo il numero di percorsi diversi lunghi quel numero di mosse. Dato che la risposta può essere molto grande, devi calcolarla modulo $10^9 + 7$, usando le funzioni di libreria fornite dal grader.

Se non è possibile raggiungere la casella di destinazione, stampa la riga "0 0".

Libreria

Il grader fornisce una libreria contente le seguenti funzioni, che permettono di effettuare operazioni aritmetiche modulo $10^9 + 7$. In tutti i casi, l'input può essere un qualunque valore `int` valido e l'output è sempre compreso nel range $0, 1, 2, \ldots, 10^9 + 6$. Un'implementazione di esempio è disponibile per provare la tua soluzione; per i dettagli leggi la sezione seguente.

- `int Add(int a, int b)`: somma due numeri a e b, ritorna il risultato modulo $10^9 + 7$.
- `int Sub(int a, int b)`: sottrae b da a, poi ritorna il risultato modulo $10^9 + 7$.
- `int Mul(int a, int b)`: calcola il prodotto di a e b, poi ritorna il risultato modulo $10^9 + 7$.

- `int Div(int a, int b)`: calcola il quoziente di a diviso per $b \neq 0$ modulo $10^9 + 7$, ovvero ritorna il valore $0 \leq q < 10^9 + 7$ se e solo se `Mul(b,q)` $= a \mod (10^9 + 7)$.

Puoi supporre che le operazioni precedenti vengano eseguite in tempo costante.

Per utilizzare queste funzione devi aggiungere la riga `#include "arithmetics.h"` nella lista degli `include` della tua soluzione.

Pratica

Viene fornito il file *sample.zip* contenente il file header `arithmetics.h` e il file `arithmetics.cpp`, una implementazione d'esempio delle precedenti funzioni che dovresti usare per provare la tua soluzione.

Per utilizzare la libreria, devi copiare entrambi i file nella stessa cartella del codice sorgente della tua soluzione (ovvero `chessrush.cpp`) e aggiungere la riga `#include "arithmetics.h"` nella lista degli `include` della tua soluzione.

A questo punto, compila `chessrush.cpp` assieme a `arithmetics.cpp`, ad esempio eseguendo `g++ -o chessrush arithmetics.cpp chessrush.cpp` nel terminale. Altrimenti, se stai usando un IDE che supporta i progetti, devi aggiungere manualmente tutti i tre file al tuo progetto prima di compilare la tua soluzione.

Le soluzioni per gli input di esempio si trovano nei file `output0.txt, output1.txt`. Nota bene, gli strumenti forniti e le funzioni della libreria non controllano che le tue soluzioni siano corrette.

Per sottomettere la tua soluzione, devi caricare nel sistema di gara solo il file `chessrush.cpp`.

Esempi

Input	*Output*

`8 8 5`
`P 1 2`
`R 4 8`
`Q 2 3`
`B 3 6`
`K 5 5`
	`0 0`
	`2 2`
	`2 5`
	`2 2`
	`7 393`

Assunzioni

$1 \leq Q \leq 1000$
$2 \leq C \leq 1000$
$C \leq R \leq 10^9$

Limite di tempo: 1.3 s

Limite di memoria: 64 MiB

Punteggi

Subtask	Punti	Assunzioni
1	0	Casi d'esempio
2	8	$T \in \{'P', 'R', 'Q'\}$, ovvero tutti i pezzi sono pedoni, torri o regine
3	15	$T = 'B'$ e $C, R \leq 100$
4	22	$T = 'B'$
5	5	$T = 'K'$ e $C, R \leq 100$ e $Q \leq 50$
6	8	$T = 'K'$ e $C, R \leq 100$
7	15	$T = 'K'$ e $C \leq 100$
8	20	$T = 'K'$
9	7	Nessuna limitazione aggiuntiva