

Turbošach (Chess Rush)

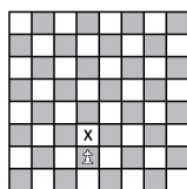
Šachové kráľovstvo má tvar obdĺžnika s R riadkami a C stĺpcami, pričom $R \geq C$. Riadky sú očíslované od 1 po R (zdola hore) a stĺpce od 1 po C (zľava doprava).

Obyvateľov šachového kráľovstva voláme *figúrky*. Je len päť typov figúrok: pešiaci (pawns), veže (rooks), strelci (bishops), dámy (queens) a králi (kings). Organizátori CEOI sú amatéri na vás dobrí, takže v tomto šachovom kráľovstve nežijú žiadni jazdci.

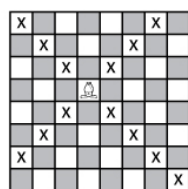
V nasledujúcom zozname pre každú figúrku uvádzame ako vyzerá každý jeden jej ťah.

- Pešiak sa môže posunúť o jeden riadok dopredu (z riadku r na riadok $r + 1$, bez zmeny stĺpca).
- Veža sa môže posunúť o ľubovoľne veľa políček buď v riadku, v ktorom práve stojí, alebo v stĺpci, v ktorom práve stojí.
- Strelec sa môže posunúť o ľubovoľne veľa políček po jednej alebo druhej uhlopriečke, na ktorej práve stojí.
- Dáma sa môže posunúť na ľubovoľné políčko, na ktoré by sa vedela dostať veža alebo strelec.
- Kráľ sa môže posunúť na ľubovoľné z (nanajvýš) ôsmich políček, ktoré s jeho aktuálnym susedia stranou alebo rohom.

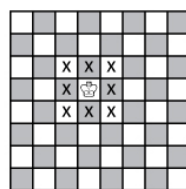
Na nasledujúcom obrázku sú pre každú figúrku písmenom X označené tie políčka, na ktoré sa vie jedným krokom dostať z políčka, kde práve stojí.



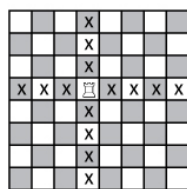
pešiak



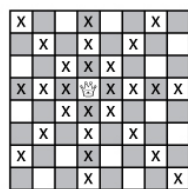
strelec



kráľ



veža



dáma

V poslednom čase sa stáva, že figúrky cestujúce po kráľovstve občas niečo zožerie a zmiznú bez stopy. Ostatné figúrky sa preto boja cestovať kráľovstvom a radi by ním čo najrýchlejšie

prebehli. Vašou úlohou bude pomôcť figúrkam zodpovedať Q otázok úzko súvisiacich s takouto formou cestovania.

V každej otázke dostanete zadanú figúrku, ktorá ide cestovať, číslo stĺpca v riadku 1 kde jej cesta začína a číslo stĺpca v riadku R , kde chce figúrka svoju cestu ukončiť. Zistite:

- Aký najmenší počet ťahov figúrka potrebuje spraviť, aby dosiahla cieľ svojej cesty?
- Koľko existuje rôznych postupností ťahov, ktoré majú správny začiatok, správny koniec aj optimálny počet ťahov? (Figúrky sú presvedčené, že čím viac optimálnych ciest existuje, tým väčšiu šancu majú vyhnúť sa cestou nebezpečenstvu, ktoré by im mohlo hroziť.)

Input

V prvom riadku vstupu sú tri celé čísla R , C a Q : rozmery šachovnice a počet otázok.

Zvyšok vstupu tvorí Q riadkov, každý z nich popisuje jednu otázku. Popis každej otázky má tvar „ $T\ c_1\ c_R$ “, kde:

- T je znak udávajúci typ figúrky ('P'=pawn=pešiak, 'R'=rook=veža, 'B'=bishop=strelec, 'Q'=queen=dáma, 'K'=king=kráľ).
- Celé čísla c_1 a c_R ($1 \leq c_1, c_R \leq C$) sú čísla stĺpcov kde figúrka v riadku 1 začína a kde chce v riadku R skončiť svoju cestu.

Output

Pre i -tu otázku vypíšete jeden riadok tvaru „ $m_i\ s_i$ “, kde m_i je minimálny počet ťahov a s_i je počet optimálnych ciest, modulo $10^9 + 7$.

Ak figúrka z i -tej otázky nevie dosiahnuť želaný cieľ cesty, vypíšete riadok „0 0“.

Library

Výpočty modulo $10^9 + 7$ si môžete robiť sami, alebo na to môžete využiť knižnicu pripravenú organizátormi. Túto knižnicu už poznáte z prípravného dňa.

Vstupy pre všetky jej metódy sú ľubovoľné `int`y. Výstupy všetkých metód sú `int`y z rozsahu od 0 po $10^9 + 6$, vrátane,

Aby ste si mohli svoje riešenie lokálne testovať, máte k dispozícii nie len hlavičkový súbor, ale aj ukážkovú implementáciu takejto knižnice.

Knižnica implementuje nasledujúce funkcie:

- `int Add(int a, int b)`: vráti $(a + b)$ modulo $10^9 + 7$.
- `int Sub(int a, int b)`: vráti $(a - b)$ modulo $10^9 + 7$.

- `int Mul(int a, int b)`: vráti $(a \cdot b)$ modulo $10^9 + 7$.
- `int Div(int a, int b)`: za predpokladu, že $b \neq 0$, vydolí hodnotu a hodnotou b , teda vráti takú hodnotu c , pre ktorú $Mul(b, c) = (a \text{ modulo } 10^9 + 7)$.

Môžete predpokladať, že oficiálna knižnica vykoná všetky tieto operácie v konštantnom čase.

Practice

Dostanete súbor *sample.zip*. V ňom nájdete hlavičkový súbor `arithmetics.h` s deklaráciou knižničných funkcií a súbor `arithmetics.cpp` obsahujúci ukážkovú implementáciu tejto knižnice.

Ak chcete túto knižnicu využiť vo svojom riešení, postupujte nasledovne:

Do vášho riešenia (`chessrush.cpp`) pridajte riadok `#include "arithmetics.h"`.

Súbory `arithmetics.*` umiestnite do toho istého adresára ako `chessrush.cpp`.

Svoje riešenie skompilujte spolu s ukážkovou implementáciou knižnice, napr. príkazom `g++ -o chessrush arithmetics.cpp chessrush.cpp` v príkazovom riadku.

Ak používate IDE, ktoré podporuje projekty, ručne pridajte všetky tri súbory do toho istého projektu.

Súbory `output0.txt` a `output1.txt` obsahujú správne výstupy pre ukážkové vstupy. Ak chcete vedieť, či váš program dáva na príkladoch správne výstupy, môžete si tie vaše porovnať s týmito.

Na testovanie odovzdávate len súbor obsahujúci vaše riešenie (súbor `chessrush.cpp`).

Examples

<i>Input</i>	<i>Output</i>
8 8 5	0 0
P 1 2	2 2
R 4 8	2 5
Q 2 3	2 2
B 3 6	7 393
K 5 5	

Constraints

$$1 \leq Q \leq 1000$$

$$2 \leq C \leq 1000$$

$$C \leq R \leq 10^9$$

Time limit: 1.3 s

Memory limit: 64 MiB

Grading

Subtask	Points	Constraints
1	0	sample
2	8	len pešiaci, veže a dámy
3	15	len strelci a $C, R \leq 100$
4	22	len strelci
5	5	len králi a $C, R \leq 100$ a $Q \leq 50$
6	8	len králi a $C, R \leq 100$
7	15	len králi a $C \leq 100$
8	20	len králi
9	7	bez ďalších obmedzení