

체스 러쉬

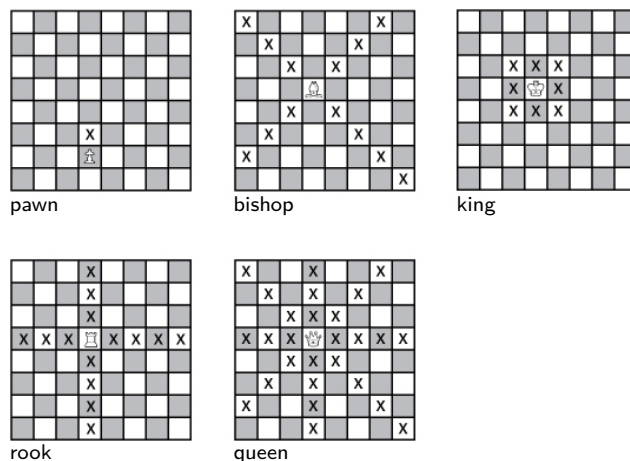
체스 랜드의 신화 세계는 R 행과 C 열이있는 사각형의 사각형 격자입니다. R 는 C 보다 크거나 같습니다. 행과 열은 각각 1 에서 R , 1 에서 C 까지 번호가 지정됩니다.

체스 랜드의 주민들은 일반적으로 일상 언어로 말로 언급되며, 졸, 루크, 비숍, 퀸, 킹과 같이 땅을 돌아 다니는 5 의 특정 유형이 있습니다. 대중의 믿음과는 달리, 체스 랜드에서는 기사도가 오래전에 죽었 기 때문에 기사를 찾을 수 없습니다.

각 조각은 정사각형에서 정사각형으로 이동하는 방식에서 고유합니다,

- 폰은 열을 변경하지 않고 한 행 앞으로 (즉, r 행에서 $r + 1$ 행으로) 이동할 수 있습니다;
- 루크는 행을 변경하지 않고 원하는 수의 열을 왼쪽 / 오른쪽으로 이동하거나 열을 변경하지 않고 원하는 수의 행을 앞 / 뒤로 이동할 수 있습니다;
- 비숍은 현재 점유 된 사각형에서 교차하는 두 대각선의 사각형으로 이동할 수 있습니다;
- 여왕은 루크 또는 비숍이 자신의 위치에서 이동할 수 있는 모든 광장으로 이동할 수 있습니다;
- 킹은 8의 인접한 사각형으로 이동할 수 있습니다.

다음 그림에서 각 조각이 한 단계로 이동할 수 있는 사각형을 X로 표시했습니다 (여기서 행은 아래에서 위로, 열은 왼쪽에서 오른쪽으로 번호가 매겨집니다).



최근 체스 랜드는 위험한 곳이되었습니다. 땅을 통과하는 조각들은 알 수없는 힘에 의해 예기치 않게 포착되어 사라질 수 있습니다. 결과적으로 그들은 가능한 한 빨리 목적지에 도달하기를 원하며 (즉, 적은 이동으로) 최소한의 단계를 사용하여 목적지에 도달 할 수 있는 다양한 방법에 관심이 있습니다. -더 많은 경로를 사용할 수 있다는 것은 점령 가능성

이 낮아질 수 있기 때문입니다. 적어도 하나의 방문한 광장에서 다른 두 경로는 다른 것으로 간주됩니다.

이 문제를 위해 말들이 1 행의 주어진 열에서 체스 랜드로 들어가고 R 행의 주어진 열에서 땅을 빠져 나간다고 가정 해 봅시다. 당신의 임무는 Q 질문에 답하는 것입니다: 조각의 유형, 1 행에 들어가는 열, 그리고 나가기 위해 R 행에 도달해야 하는 열이 주어지면 최소 이동 횟수를 계산합니다. 체스 랜드에서 할 수 있는 다양한 방법이 있습니다.

입력

첫 번째 줄에는 공백으로 구분 된 3 개의 정수 R, C 및 Q , 체스 랜드의 행 및 열 수, 질문 수가 각각 포함됩니다. 그런 다음 Q 줄이 이어집니다.

각 라인은

- 문제의 조각 유형에 해당하는 T 문자 (폰은 'P', 루크는 'R', 비숍은 'B', 퀸은 'Q', 킹은 'K')
- 두 개의 정수 c_1 및 c_R , $1 \leq c_1, c_R \leq C$, 조각이 1 행의 c_1 번째 열에서 시작하여 R 행의 c_R 번째 열에 도달해야 함을 나타냅니다.

산출

Q 줄을 인쇄해야 합니다. i -번째 줄에는 두 개의 공백으로 구분 된 정수가 들어 있습니다. i -번째 질문에 대한 답입니다. 첫 번째 줄은 필요한 최소 단계 수이고 두 번째 줄은 이 단계 수를 사용하여 사용할 수 있는 다른 경로. 답이 상당히 클 수 있으므로 그레이더가 제공하는 라이브러리 함수를 사용하여 $10^9 + 7$ 모듈로 계산해야 합니다.

목표 사각형에 도달 할 수 없는 경우 라인 "0 0" 을 출력합니다.

도서관

채점자는 기본 산술 모듈로 $10^9 + 7$ 와 관련된 계산을 수행하기 위해 다음 라이브러리 함수를 제공합니다. 모든 경우에 입력은 유효한 int 값이 될 수 있으며 출력 범위는 $0, 1, 2, \dots, 10^9 + 6$ 입니다. 솔루션 테스트에 샘플 구현을 사용할 수 있습니다. 자세한 내용은 다음 섹션을 참조하십시오.

- `int Add(int a, int b)`: 숫자 a 및 b 를 더하고 $10^9 + 7$ 모듈로 답을 반환합니다.
- `int Sub(int a, int b)`: a 에서 b 를 뺀 다음 $10^9 + 7$ 모듈로 답을 반환합니다.
- `int Mul(int a, int b)`: a 와 b 의 곱을 계산 한 다음 $10^9 + 7$ 모듈로 답을 반환합니다.
- `int Div(int a, int b)`: a 의 몫을 $b \neq 0$ 모듈로 $10^9 + 7$ 로 나눈 값을 계산합니다, i.e. 값을 반환 $0 \leq q < 10^9 + 7$ 경우에만 $Mul(b, q) = a \mod (10^9 + 7)$.

위의 모든 작업이 일정한 시간에 수행된다고 가정 할 수 있습니다.

이러한 함수에 액세스하려면 솔루션의 포함 목록에 `#include "arithmetics.h"` 줄을 추가해야 합니다.

연습

보충 파일 *sample.zip*에는 솔루션 테스트에 사용해야 하는 이전 함수의 구현 예인 `arithmetics.h` 헤더와 `arithmetics.cpp` 파일이 포함되어 있습니다. .

이를 활용하려면 두 파일을 솔루션 소스 (예: `chessrush.cpp`) 가있는 동일한 디렉토리에 복사하고 `#include "arithmetics.h"` 줄을 해당 파일에 추가해야 합니다. 목록을 포함하십시오.

그런 다음 `chessrush.cpp` 를 `arithmetics.cpp` 와 함께 컴파일하십시오. 명령 줄에서 `g++ -o chessrush arithmetics.cpp chessrush.cpp` 를 사용합니다. 또는 프로젝트 기반 IDE를 사용하는 경우 솔루션을 빌드하기 전에 이러한 세 파일을 모두 프로젝트에 수동으로 추가해야 합니다.

샘플 입력에 대한 정답은 `output0.txt`, `output1.txt` 파일에서 찾을 수 있습니다. 제공된 도구 및 기능 중 어떤 것도 답변의 정확성을 확인하지 않습니다.

제출할 때는 `chessrush.cpp` 파일 만 채점 시스템에 업로드해야 합니다.

예

입력	산출
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	

제약

$$1 \leq Q \leq 1000$$

$$2 \leq C \leq 1000$$

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

시간 제한: 1.3 s

메모리 제한: 64 MiB

채점

하위 작업	포인트들	제약
1	0	견본
2	8	$T \in \{'P', 'R', 'Q'\}$, i.e. 모든 조각은 폰, 루크 또는 퀴입니다.
3	15	$T = 'B'$ 과 $C, R \leq 100$
4	22	$T = 'B'$
5	5	$T = 'K'$ 과 $C, R \leq 100$ 과 $Q \leq 50$
6	8	$T = 'K'$ 과 $C, R \leq 100$
7	15	$T = 'K'$ 과 $C \leq 100$
8	20	$T = 'K'$
9	7	추가 제약 없음