

XIV BXComp

14º Campeonato de Programação para Calouros do Curso de Sistemas de Informação
2024

0ª Etapa - 6 pontos

Os robôs de Jimmy

Retroville está mais agitada do que nunca, especialmente porque Jimmy Neutron, o jovem gênio, decidiu automatizar praticamente todas as tarefas na cidade com seus novos robôs superinteligentes. Desde cortar a grama do quintal da Vovó Neutron até entregar pizzas na casa do Carl, os robôs de Jimmy estão por toda parte. Mas, como todas as invenções de Jimmy, nem tudo sai conforme o planejado.

Recentemente, durante uma demonstração de suas criações para a classe, algo hilário aconteceu: dois robôs entregadores de pacotes tentaram atravessar a mesma porta ao mesmo tempo e ficaram presos, provocando uma avalanche de pacotes de presente para todos os lados. Miss Fowl, com um pacote de sucrilhos na cabeça, não achou graça. Jimmy, rindo nervosamente, percebeu que precisava resolver esse problema antes que os robôs causassem mais confusão.

Agora, Jimmy precisa da sua ajuda para garantir que seus robôs possam completar suas tarefas sem colidir uns com os outros, especialmente porque Hugh Neutron começou a usar os robôs para suas competições de panqueca (não perguntem!). Cada robô começa em uma posição específica e segue uma sequência de comandos. Seu trabalho é verificar se esses robôs podem completar suas tarefas sem se chocarem e causarem mais desastres cômicos.



Figura 1: Jimmy Neutron

Tarefa

Você deve escrever um programa que verifique se todos os robôs completarão suas sequências de comandos em uma grade de movimentação sem colidir uns com os outros. Considere que todos os robôs se movimentam ao mesmo tempo, isto é, todos os robôs fazem seu primeiro movimento ao mesmo tempo e seguem assim até completarem a sequência.

Entrada

A primeira linha da entrada consiste de um inteiro T que representa a quantidade de casos de teste.

Para cada um dos T ($1 \leq T \leq 100$) casos de teste, as entradas se dão como se segue:

Uma linha contendo dois inteiros N e M ($1 \leq N, M \leq 100$), as dimensões da grade de tamanho N verticalmente e M horizontalmente.

Uma linha contendo dois inteiros R e K ($1 \leq R, K \leq N \times M$), respectivamente, o número de robôs e o número de movimentos de cada robô.

As próximas R linhas contêm dois inteiros x_i, y_i ($0 \leq x_i < N, 0 \leq y_i < M$) e uma string S_i de K caracteres, representando a posição inicial (x_i, y_i) e a sequência de movimentos S_i do i -ésimo robô.

A string S_i é composta pelas letras 'E', 'D', 'C' e 'B', representando, respectivamente, os movimentos para esquerda, direita, cima e baixo.

É garantido que os robôs começam todos em posições distintas e que toda sua sequência de movimento permanece dentro da grade de movimentação. Cada movimento dos robôs move exatamente uma unidade para a direção escolhida.

A posição (0, 0) representa um robô na extremidade superior esquerda da grade.

Saída

Imprima T linhas, representando a resposta para cada caso de teste.

Imprima "Eureka! Eu sabia que minha programação era impecável." caso os robôs completem sua sequência de comandos sem colisões e "Hmm, parece que tivemos um pequeno contratempo robótico." caso os robôs tenham alguma colisão.

Exemplo de Entrada

```
2
4 4
3 2
0 0 DB
3 3 CB
1 1 ED
5 5
3 4
0 0 BDDD
1 0 DCDD
4 0 CCCC
```

Exemplo de Saída

```
Hmm, parece que tivemos um pequeno contratempo robótico.
Eureka! Eu sabia que minha programação era impecável.
```