

(REV_05) Problema: Em diversos jogos de vídeo game o jogador é desafiado a solucionar *puzzles* para atravessar um determinado local no mapa. Sendo assim, imagine que você se deparou com o desafio de atravessar um galpão de acordo com um documento de instruções. O mapa do galpão é representado por uma matriz de M linhas e N colunas. Cada posição dessa matriz assume o valor 0 (livre) ou 1 (bloqueado). **O jogador só pode se movimentar por posições livres.** No documento de instruções consta a posição inicial do jogador (X_i, Y_i) e a posição final desejada (X_f, Y_f) . Além disso, a instrução contém uma sequência de quatro caracteres que representa a prioridade de movimentação do jogador: a(C)ima, a(B)aixo, (D)ireita, (E)squerda. **O jogador nunca se movimenta pela diagonal e nunca pode voltar para uma posição que ele já visitou.**

Faça um programa que leia todas essas informações e movimente o jogador pelo mapa procurando uma saída. Seu programa deve registrar todas as posições movimentadas até chegar a posição final, se for possível.

Exemplo de funcionamento

Na Figura 1 é ilustrado um exemplo de funcionamento para o programa. Na Figura (a), o mapa é representado por uma matriz 7 x 7. As posições em verde (0) estão livres e as vermelhas (1) bloqueadas. A posição de início é a (1, 3) e a final (7, 3). A prioridade de movimentação é BDEC, ou seja, o jogador deve tentar se movimentar na seguinte ordem: abaixo, direita, esquerda ou acima. Na Figura (b) é ilustrada o caminho percorrido pelo jogador da posição inicial até a final. A saída deste programa deve ser:

(1,3) (2,3) (2,4) (3,4) (4,4) (4,5) (5,5) (6,5) (6,4) (6,3) (7,3)

Se não existir um caminho viável até a posição final, a saída do programa deve ser todas as posições visitadas até ele parar. Por exemplo, imagine que a posição (6,4) deste exemplo esteja bloqueada. Neste caso, devido às regras de movimentação, não existe caminho para alcançar o ponto final. Sendo assim, a saída deve ser:

(1,3) (2,3) (2,4) (3,4) (4,4) (4,5) (5,5) (6,5)

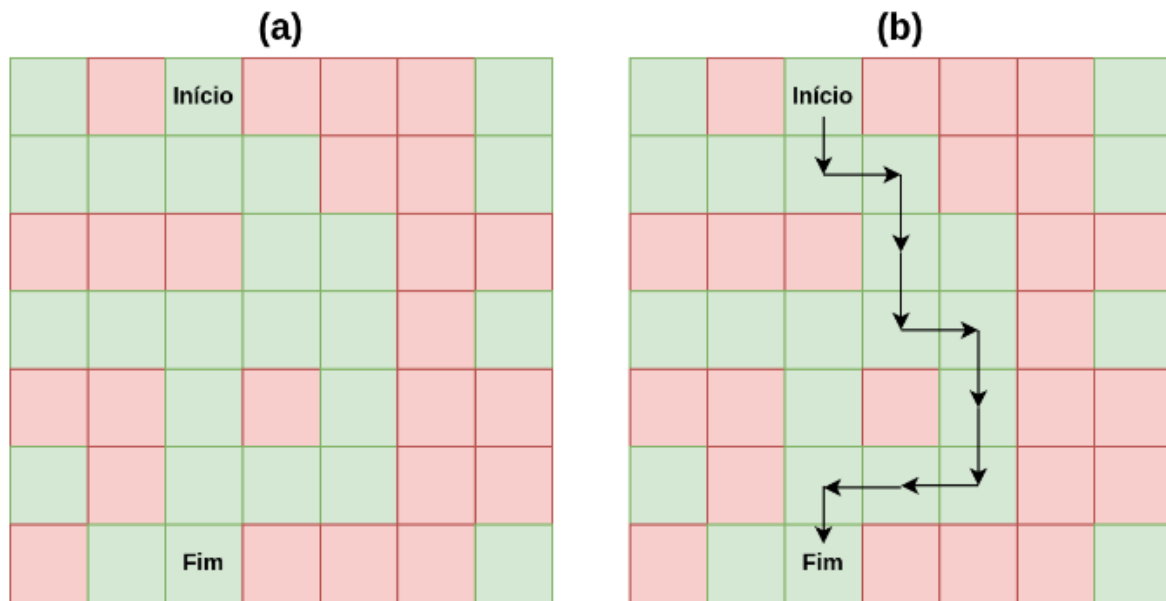


Figura 1: Exemplo de mapa e de movimentação do jogador de acordo com a prioridade BDEC

Definição dos formatos de entrada e saída:

- Entrada: Valores M e N definindo a quantidade de linhas e colunas da matriz, acompanhado da matriz em si. Em seguida, será informado a posição inicial do jogador (X_i, X_j) e a posição final desejada (X_f, Y_f) . Por último, será dado uma sequência de quatro caracteres que representa a prioridade de movimentação do jogador.
- Saída: Seu programa deve retornar todos os pontos visitados até a saída, se for possível. Se não for possível, ele retorna até onde foi possível.

Ver exemplos de formato de entrada e saída nos arquivos fornecidos com a questão.