

7ª exercício avaliativo (EA7)
Programação II (INF16153) - UFES
13 de Julho de 2023

Em diversos jogos de vídeo game o jogador é desafiado a solucionar *puzzles* para atravessar um determinado local no mapa. A ideia deste EA é simular um destes *puzzles*.

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 , X_j) 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. Veja os casos de teste para compreender melhor o padrão de entrada e saída.

Padrão de entrada

Seu programa deve ser capaz de ler o seguinte padrão de entrada

```
a11 a12 a13 ... a1n (primeira linha de uma matriz com N colunas)
...
am1 am2 am3 ... amn (última linha de uma matriz com N colunas)
F (caractere que indica o fim da leitura da matriz)
Xi Yi (ponto inicial)
Xf Yf (ponto final)
BCDE (prioridade de movimentação neste formato (este é apenas um exemplo))
```

Padrão de 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. Exemplo:

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

Veja o exemplo ilustrado a seguir para entender melhor.

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 7×7 . 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: **a**baixo, **d**ireita, **e**squerda ou **a**cima. 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)

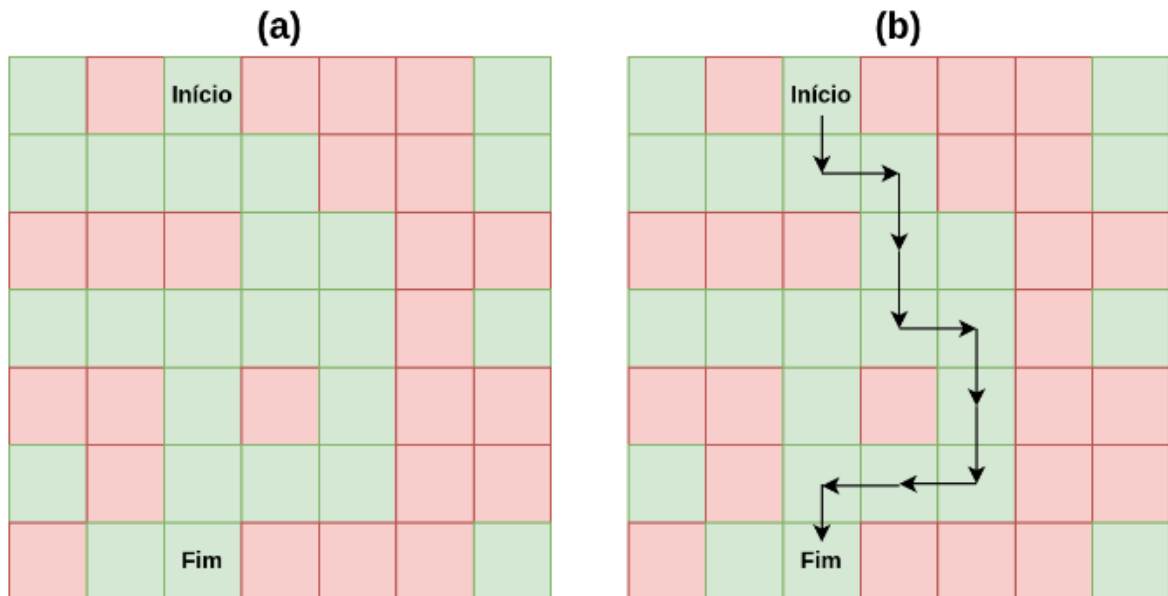


Figura 1: Exemplo de mapa e de movimentação do jogador de acordo com a prioridade BDEC

Regras gerais

- Deve ser utilizando entrada e saídas padrões do C
- A atividade é **individual**. Todas as questões serão testadas e plágio não será tolerado
- Seu programa deve implementar um TAD para labirinto com **encapsulamento total (opaco)**
- Toda lógica de negócio deve ser implementada preferencialmente fora da sua `main()`. Em outras palavras, você só deve fazer chamadas de funções do seu TAD e/ou bibliotecas (como realizado em exercícios na sala de aula)
- Seu programa será testado com o Valgrind para detectar vazamento de memória e erros de alocação. É sua responsabilidade liberar toda memória alocada.
- **Números de ponto flutuante devem ter precisão simples e apenas duas casas decimais devem ser impressas**
- Você deve fornecer um **Makefile** que gere um arquivo executável chamado **EA7**
- **O seu programa será executado da seguinte forma:**
 - `./EA7 < entrada > saída`
- Haverá **correção automática**, portanto, siga os padrões de saída corretamente
 - O corretor ignora espaços e quebras de linha
 - Porém, se você escrever informações na tela, ele retornará um erro na saída (por exemplo: "digite uma carta"). Portanto, escreva somente o que foi solicitado na tela
- Organização, modularização e boas práticas de programação são critérios fundamentais de avaliação.
- A submissão da atividade será realizada via Github de acordo com as instruções já conhecidas
 - Criar uma pasta EA7 e submeter a atividade dentro dela

