

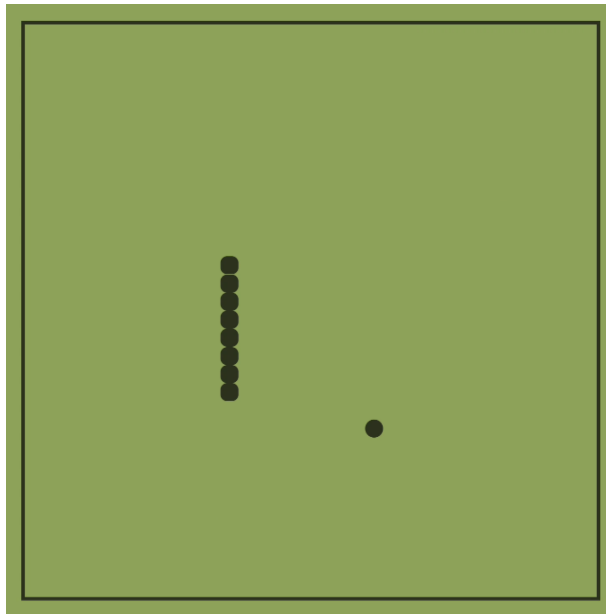
MC202AB - Estrutura de Dados

Lab 06 - Variações de Listas

Data da Primeira Chance: 10 de Outubro de 2022

Peso: 3

O jogo Snake, também conhecido como "jogo da cobrinha", é um jogo antigo que ficou conhecido por diversas versões, cuja versão inicial começou com o jogo Blockade de 1976.



A cobra se move numa matriz (MxN) de posições (x,y) e seu objetivo é se mover por essa matriz e se alimentar com as frutas que aparecem aleatoriamente em posições diferentes (elas sempre aparecem individualmente).

Ela começa com tamanho 1 e a cada vez que se alimenta, seu tamanho cresce e uma nova fruta aparece.

O jogo deve **parar** assim que ocorrer:

- A colisão da cobra com seu próprio corpo (GAME OVER);
- A cobra ocupar todas as posições (YOU WIN);

Você deve primeiro checar se a cobra está na posição para onde a cabeça irá e só mover a cobra depois.

O usuário controla a direção da cabeça da cobra (**para cima, para baixo, esquerda e direita**) e seu corpo segue.

- **'w'**: a cobra se movimenta para cima, da posição (x,y), para (x-1,y);

- 's': a cobra se movimenta para baixo, da posição (x,y), para (x+1,y);
- 'a': a cobra se movimenta para a esquerda, da posição (x,y), para (x,y-1);
- 'd': a cobra se movimenta para direita, da posição (x,y), para (x,y+1);

Implemente o jogo da cobrinha utilizando **Lista Duplamente Encadeada**, que armazena a posição (x,y) da matriz do jogo que ela ocupa (cada ponto no corpo da cobra).

- A cabeça da lista é a cabeça da cobra;
- A cauda da lista é a cauda da cobra;
- Quando a cobra come a fruta, sua cauda é acrescentada ao seu tamanho;

Quando a cobra passa por pelo limite horizontal ela aparece no começo da matriz, por exemplo a posição (0,N-1), se seu próximo movimento for para DIREITA, sua cabeça teria posição (0,0). O mesmo vale para os outros limites.

Lembrando que a cobra deve fazer as operações e seguir as regras do jogo.

A matriz possui valor zero em todas as posições, exceto:

- Os lugares que a cobra ocupa possuem valor 1;
- O lugar que a fruta ocupa no espaço de tempo em que ela está em uma certa posição, possui valor -1.

Entrada

A entrada é formado por:

- Inteiro **M** - Tamanho das linhas da Matriz do jogo, onde $M \geq 2$;
- Inteiro **N** - Tamanho das colunas da Matriz do jogo, onde $N \geq 2$;
- **Posição (x,y)** da fruta no jogo no começo e a cada vez que a fruta é comida pela cobra (É garantido que ela nunca cairá numa posição em que a cobra ocupa): *FRUTA x y*
- **Posição (a,b)** inicial da cobra no jogo, sempre será dada na 3 linha da entrada, apenas no começo do jogo: *COBRA a b*
- **Sequência de movimentos (Ex: w, s, a, d)** dos movimentos a serem feitos pela cobra, até ela comer a fruta;

As entradas são feitas para dar fim de jogo na última jogada.

```
[M N]
FRUTA [X Y]
COBRA [A B]
[MOVIMENTO]... (Até a cobra comer a fruta)
FRUTA [X Y] ... (Até o fim do jogo)
```

Saída

Impressão da matriz do jogo inicial (com a posição inicial da fruta e da cobra) e a cada movimento da cobra e inserção da fruta com um **espaçamento de uma linha** por impressão.

A matriz terá as seguintes características:

- Cada caractere da matriz terá um espaço “ ”;

- Nos lugares onde a cobra está, será impresso o caractere ‘#’
- Nos lugares onde a fruta está, será impresso o caractere ‘*’
- Nos demais lugares da matriz do jogo, será impresso o caractere ‘_’

Quando houver fim de jogo, deve-se imprimir o último estado da matriz do jogo e em seguida:

- Se houver fim de jogo com a cobra batendo em seu próprio corpo, imprimir a frase **“GAME OVER”** ou,
- Se a cobra ocupar todas as posições da matriz imprimir **“YOU WIN”**.

```
[MATRIZ DO JOGO MxN]
[\n]
...
[ÚLTIMO ESTADO DA MATRIZ DO JOGO MxN]
[\n]
[FIM DE JOGO, YOU WIN OU GAME OVER]
```

Exemplos

Exemplo 1:

Entrada

```
2 2
FRUTA 0 0
COBRA 1 1
w
a
FRUTA 1 1
s
d
FRUTA 0 1
w
```

Saída

```
* _
_ #
* #
_ _
# #
_ _
```

```
# #
_ *

# _
# *

# _
# #

# *
# #

# #
# #

YOU WIN
```

Exemplo 2:

Entrada

```
3 3
FRUTA 1 2
COBRA 2 2
w
FRUTA 2 1
a
s
FRUTA 2 2
d
FRUTA 0 0
d
w
w
FRUTA 1 2
d
d
s
FRUTA 1 1
a
FRUTA 2 1
w
```

Saída

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

— — —
— — —
— — —

```

# # #
#  _ *
#  _ _

# # #
#  _ #
#  _ _

# # #
# * #
#  _ _

# # #
# # #
#  _ _

# # #
# # #
# * _

# # #
# # #
# * _

```

GAME OVER

Regras e Avaliação

O aluno é obrigado a utilizar **lista duplamente encadeada** para as posições da cobra na matriz do jogo.

Seu código será avaliado não apenas pelos testes do CodePost, mas também pela qualidade. Dentre os critérios subjetivos de qualidade de código iremos analisar: o uso apropriado de funções, de comentários, e de structs; a escolha de bons nomes de funções, variáveis e de structs e seus campos; o correto uso de Tipos Abstratos de Dados e a correta separação em vários arquivos; a ausência de diversos trechos de código repetidos, e o tempo de execução e uso de memória dos algoritmos projetados. Note, porém, que essa não é uma lista exaustiva, pois outros critérios podem ser analisados dependendo do código apresentado visando mostrar ao aluno como o código poderia ser melhor.

Submissão

Você deverá submeter no CodePost, na tarefa Lab 6, um arquivo com o nome `lab06.c`, um arquivo `cobra.h` e um arquivo `cobra.c`. Após o prazo estabelecido para a atividade, será aberta uma tarefa Lab 6 - Segunda Chance, com prazo de entrega até o fim do semestre.