Universidade Federal do Espírito Santo – Centro Tecnológico
Departamento de Informática
Prof. Vinícius Fernandes Soares Mota

UFES

1. EA – Batalha Naval

ATENÇÃO: Para a devida entrega desse exercício, pedimos que você disponibilize o link do seu código do replit, no ava. (Caso queiram correções com comentários em seus códigos do replit adicione o seguinte email como convidado: guisgb10@gmail.com)

Batalha naval é um jogo de tabuleiro em que dois jogadores posicionam navios de diferentes tamanhos dentro de quadros e tentam adivinhar a posição dos navios do quadro adversário em cada rodada. Neste exercício avaliativo, há a simplificação do jogo batalha naval, implementando apenas o tabuleiro de um dos jogadores com os navios já previamente posicionados, fornecidos por meio de arquivos.

Implemente um programa que faça a leitura do arquivo navios.dat, que contém a posição dos navios em um tabuleiro de 10 linhas por 10 colunas. O navios.dat contém uma sequência de pares de números inteiros que representam a posição dos navios.

Exemplo de navios.dat após realizar a leitura:

1 5 1 6

(indica a linha e coluna de cada peça do navio que será posicionada, neste caso na linha 1 coluna 5 há uma peça de navio e na linha 1 coluna 6 há outra peça de navio)

Em seguida, o programa implementado deverá armazenar o tabuleiro e a posição dos navios por meio de matrizes dinâmicas. As matrizes, juntamente com os navios, deverão ser apresentadas na tela conforme o exemplo abaixo:

```
- - - - o o - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
```

Por meio de entrada no teclado, o jogador deverá informar a posição de destruição do navio. Caso erre a posição, não haverá nenhuma modificação na imagem do jogo. Caso acerte a posição, o local do navio será representado por um X conforme o exemplo abaixo:

UFES

```
- - - - x o - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
```

Quando não houver mais navios para destruir, o programa deverá sair, liberando toda a memória utilizada.

OBS: o programa não deverá apresentar vazamentos de memória. Toda variável deve utilizar alocação dinâmica.

UFES

Legenda:
x - Representa a parte do navio destruída
o - Representa a parte do navio não destruída
- - Representa um local em que não há navios

Exemplo 2: **1,5, 1,6, 3,4, 3,5, 3,6, 3,7, 3,8, 4,3, 5,3, 6,3, 7,3, 6,7, 7,7, 8,7**

```
- - - - o o - - - -
- - - - - - - - - -
- - - o o o o o - -
- - o - - - - - - -
- - o - - - - - - -
- - o - - - o - - -
- - o - - - o - - -
- - - - - - o - - -
- - - - - - - - - -
- - - - - - - - - -
```

Exemplo 3: **1,1, 2,1, 5,5, 5,6, 5,7, 5,8, 10,10, 10,9, 10,8, 10,7, 9,10, 8,10, 7,10, 6,10**

```
o - - - - - - - - -
o - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - o o o o - -
- - - - - - - - - o
- - - - - - - - - o
- - - - - - - - - o
- - - - - - - - - o
- - - - - - o o o o
```