

Trabalho em Grupo

Este é um trabalho em **grupo de 3 alunos**; o trabalho consiste da criação de um programa que resolva o problema especificado abaixo. Procure parceiros; leia com atenção as instruções abaixo e bom trabalho.

1 O problema

Criar um jogo similar ao Batalha Naval. Nesse jogo, uma frota está escondida no oceano, portanto, você não consegue perceber onde estão as naves que compõem a frota. O objetivo do jogo é você destruir a frota, isto é, afundar todas as naves com a quantidade de torpedos disponível.

1 O Oceano

O Oceano é representado por uma matriz de ordem O; a Figura 1 apresenta um Oceano de ordem 5; cada célula da matriz é um Setor do Oceano.

1.2 As Naves

Há dois tipos de naves: destruidor e submarino. Um destruidor ocupa dois setores do Oceano nos sentidos horizontal ou vertical; um submarino ocupa apenas um setor do Oceano.

As naves podem ser dispostas em qualquer parte do Oceano, contudo, duas naves não podem estar em setores contíguos no Oceano; sempre deve haver um setor vazio entre elas.

Há uma quantidade D de destruidores e S de submarinos.

1.3 Jogabilidade

O jogo inicia apresentando o Oceano na tela do computador e a quantidade de torpedos disponível para o jogador; a frota está distribuída no Oceano, mas não é percebida visualmente pelo jogador. Em seguida, o jogo dá oportunidade ao jogador para o lançamento de um torpedo; um lançamento é representado pela digitação de um setor, no qual o jogador supõe que haja uma nave ou parte de uma. Por exemplo, o jogador digita C3 e essa digitação representa o lançamento de um torpedo para o setor C3 do Oceano.

Após cada lançamento, o jogo informa se o torpedo:

1. acertou parte de uma nave;
2. afundou uma nave;
3. se perdeu na água.

	1	2	3	4	5
A					
B					
C					
D					
E					

Figura 1. O oceano.

Nos casos 1 e 2, a informação de que o torpedo acertou algo é representada por uma nova apresentação do Oceano na tela do computador, mas agora com a visualização do efeito do torpedeamento. Esse efeito é a apresentação de um asterisco (*). Além disso, o jogo deve informar textualmente se o torpedo afundou um submarino ou acertou/afundou um destruidor. No caso 3, o jogo simplesmente escreve a letra 'W' na tela computador.

A Figura 2 mostra uma visualização do Oceano após o jogador ter lançado 2 torpedos. O lançamento apontado para o setor B4 se perdeu na água, representado pela letra 'W' nesse setor. Contudo, o lançamento para o setor C2 acertou uma nave, representado pelo '*' nesse setor. O jogo, então, informa que o torpedo afundou um submarino. Em seguida, o jogo apresenta textualmente a quantidade de naves que faltam para serem afundadas e a quantidade de torpedos disponível.

	1	2	3	4	5
A					
B				W	
C		*			
D					
E					

You sank a submarine.

Destroyers: 3
Submarines: 5
Torpedos: 4

Figura 2. O Oceano e uma nave torpedeada no setor C3.

Independentemente, dos casos 1, 2 ou 3, o jogo novamente dá oportunidade ao jogador para um novo lançamento de torpedo. O jogador tem à sua disposição T torpedos.

Cada vez que o jogador afunda uma nave, o jogo informa "Afundou um destruidor" ou "Afundou um submarino". O afundamento de uma nave precisa ser total, assim, afundar um destruidor demanda dois torpedos certos.

O jogo termina quando o jogador não tiver mais torpedos ou quando ele tiver destruído a frota.

2 Restrições

- O grupo deve ter realizado inscrição para realização deste trabalho.
- A linguagem de programação é C.
- O uso do padrão de codificação da disciplina é obrigatório.
- O Oceano deve ser apresentado na tela do computador tal como apresentado na Figura 1.
- A jogabilidade deve ser aquela descrita Seção 1.3.
- O = 5; D = 0; S = 3
- Antes de um lançamento de torpedo, os valores de D, S e T devem estar visíveis ao jogador.
- A posição da frota deve ser disposta de modo pseudoaleatório, considerando o espaço obrigatório entre as naves, via função rand().
- Não usar função srand().
- O jogo deve ter dois modos: PLAY e CORR.

- PLAY: nesse modo, cada apresentação do Oceano mostra apenas o efeito dos torpedos lançados. Isto é, o Oceano é apresentado com '*' ou 'W'. Esse é o modo utilizado para jogar o jogo.
 - CORR: nesse modo, o Oceano é apresentado com a frota à vista. Esse modo é o utilizado para o desenvolvimento do jogo e para a correção do trabalho.
 - O programa deve ter a variável **int MD**. Essa variável indica se o programa será jogado no modo PLAY ou CORR. Ela é obrigatória e deve estar declarada FORA da função `main()`.
- k. O valor de T e o modo do jogo (PLAY ou CORR) devem ser informados em tempo de lançamento do programa. T deve possuir o rótulo -t e o modo de jogo deve possuir o rótulo -m. Por exemplo: considerando que
- o programa executável está contido no arquivo `a.out`,
 - a quantidade de torpedos desejada é 6,
 - o modo do jogo será PLAY então o lançamento do programa deve ser:
 - ✓ `./a.out -t 6 -m PLAY`
 - ou
 - ✓ `./a.out -m PLAY -t 6`
- l. O programa deve ser organizado em funções. Pelo menos, as funções a seguir devem estar presentes no programa.
- `int argsOk (int argc, char *argv[]);`
 - ✓ `argsOk()` verifica se os argumentos passados para o programa estão consistentes com o que se espera para os mesmos;
 - ✓ Retorna
 - zero, se os argumentos estiverem consistentes;
 - valor diferente de zero, correspondente ao erro encontrado nos argumentos. O valor retornado é prerrogativa do grupo.
 - `void errorMsg (int x);`
 - ✓ `errorMsg()` apresenta em `stderr` uma mensagem de texto consistente com o valor em x. O valor em x vem do resultado retornado pela função `argsOk()`.
 - `void initOcean (char ocean[0][0]);`
 - ✓ `initOcean()` inicia o oceano com espaço (' ') em todas as posições;
 - ✓ O é uma constante que representa a quantidade de linhas e de colunas do Oceano;
 - `void showInventory (int d, int s, int t);`
 - ✓ `showInventory()` apresenta a quantidade de destruidores, submarinos e torpedos restantes para o jogador;
 - `void showOcean (const char ocean[0][0]);`
 - ✓ `showOcean()` apresenta o oceano após o lançamento de um torpedo. A apresentação deve ser consistente com o modo do jogo indicado em MD;
 - `void submarinesIntoOcean (char ocean[0][0]);`
 - ✓ `submarinesIntoOcean()` posiciona os submarinos no Oceano, conforme restrição h.

3 O que entregar

Um arquivo **.c** contendo um código-fonte, que resolva o problema deste trabalho. O nome desse arquivo deve ser **bn2-**num**.c**

num deve ser substituído pelo número de matrícula de um dos membros do grupo.

4 Como entregar

O arquivo (vide Seção3) deve ser entregue na Seção **Tarefas** do **Teams**®. Atente para a data e horário de entrega.

Atenção

- Cuide para devolver (subir) o arquivo ANTES do prazo final estabelecido no Teams; entregas atrasadas não serão aceitas.
- Basta uma devolução por grupo. Isto é, basta que UM DOS MEMBROS do grupo suba o arquivo no Teams.
- O arquivo poderá ser devolvido quantas vezes o grupo achar necessário, desde que dentro do prazo estabelecido. A cada nova devolução, você perceberá que o nome do arquivo receberá um número adicional, indicando a quantidade de vezes que o mesmo foi devolvido. É um procedimento padrão do Teams e não há com o que se preocupar com isso.