

3ª exercício avaliativo (EA3)
Programação II (INF16153) - UFES
18 de Maio de 2023

A NBA (*National Basketball Association*) é o maior e melhor torneio de basquete do mundo. Ele é dividido em duas conferências: Leste e Oeste. Cada conferência possui 15 franquias que realizam 82 jogos na temporada regular e os 8 melhores colocados de cada conferência vão para os playoffs.

O objetivo desta atividade é criar um programa (em C) para gerenciar um mini campeonato da NBA. O programa deve registrar franquias e partidas e na sequência retornar um relatório com a quantidade de vitórias, derrotas e aproveitamento de cada time da franquia e de todos os times da conferência. Para isso, considere as seguintes informações:

- Cada time da franquia possui um nome e a conferência que ele pertence
 - O nome do time sempre será maiúsculo e terá até 32 caracteres
 - O nome da conferência será sempre LESTE ou OESTE
- Uma partida de basquete nunca termina empatada e o vencedor é o time que faz mais pontos
- Uma partida sempre ocorre na arena de algum time da franquia. Neste caso, o time que recebe o adversário é considerado como jogando em casa e o adversário jogando fora de casa

Padrão de entrada

Seu programa deve ler as franquias cadastrados no torneio e, na sequência, todas as partidas realizadas até o momento. Essa ordem não será alterada.

O padrão de entrada de uma franquia é o caractere **F** seguido do nome da franquia e a conferência. Por exemplo: **F CELTICS LESTE**

Já o padrão de entrada de uma partida é representado pelo caractere **P** seguido do nome do time que joga fora de casa, o caractere **@**, time que joga em casa e pontuação de cada time. Por exemplo: **P CELTICS @ BUCKS 102 99** (se ler: Celtics jogando em Bucks e venceu por 102 a 99).

A leitura dos dados de entrada deve ser encerrada quando o programa ler o caractere **E**.

Padrão de saída

Após o encerramento da leitura dos dados de entrada seu programa deve retornar um resumo da temporada. Para cada franquia inscrita na competição, o programa deve escrever na tela (seguindo a mesma ordem de leitura): nome da franquia, abreviação da conferência ([CL] para conferência leste e [CO] para oeste), número de vitórias da franquia, número de derrotas da franquia, % de aproveitamento de vitória, número de vitórias em casa, número de vitórias fora de casa. Por exemplo:

CELTICS [CL] 5 5 50.00 3 2

Ou seja, o CELTICS da conferência leste venceu 5 partidas, perdeu 5 partidas, tem um aproveitamento de 50.00% e venceu 3 partidas em casa e 2 fora de casa.

Após exibir o resumo de todos os times, seu programa deve gerar o número de vitórias, derrotas e a % aproveitamento do acumulado das franquias de cada conferência. É simples, basta somar todas as vitórias e derrotas de todos os times da franquia e calcular o aproveitamento. Por exemplo:

LESTE 6 4 60.00

OESTE 2 2 50.00

Ou seja, as franquias da conferência leste acumulam 6 vitórias e 4 derrotas, o que dá um aproveitamento de 60.00%. Já as franquias da conferência oeste acumulam 2 vitórias e 2 derrotas, o que resulta em um aproveitamento de 50.00%. Observe que esse cálculo não é uma comparação entre as duas conferências, uma vez que os times de uma mesma conferência jogam entre eles. É apenas um resultado acumulado dos times.

Exemplos

Junto a este documento é fornecido exemplos de entrada e saída.

Regras gerais

- A atividade é **individual**. Todas as questões serão testadas e plágio não será tolerado
- Seu programa deve implementar pelo menos um TAD com **encapsulamento total (opaco)**. Em outras palavras, dados de estruturas não podem ser acessados diretamente fora do TAD
 - Se esse critério não for cumprido, sua nota será zero
- Toda lógica de negócio deve ser implementada fora da sua `main()`. Em outras palavras, você só deve fazer chamadas de funções do seu TAD (como realizado no exercício em 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 deve ter precisão simples e apenas duas casas decimais devem ser impressas**
- Você deve fornecer um **Makefile** que gere um arquivo executável chamado **EA3**
- **O seu programa será executado da seguinte forma:**
 - `./EA3 < 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 um time”). 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 EA3 e submeter a atividade dentro dela