



Universidade Federal do Espírito Santo
Departamento de Informática

1ª exercício avaliativo (EA1)
Programação II (INF16153) - UFES
30 de outubro de 2025

Campeonato Computeiro de Futsal

Diversos campeonatos de esportes realizam partidas entre times/equipes, onde cada uma possui uma pontuação que supostamente mostra o quão bom o time é em relação aos demais. Considere um campeonato de futebol de pontos corridos. Nele, duas equipes participam de partidas que necessariamente terminam ou com a vitória de alguma equipe (a que fez mais gols) ou com empate entre elas (caso ambas as equipes tenham feito o mesmo número de gols).

Cada campeonato é composto de até 20 equipes e até 200 jogadores. O número de partidas disputadas depende do número de equipes N . Considere que toda equipe jogará contra todas as outras equipes ao menos duas vezes: uma “em casa” e outra “fora de casa” (isto é, cada equipe deve jogar $2N-2$ partidas, e o total de partidas é $N^2 - N$). Com as partidas, as equipes conquistam pontos conforme a seguinte regra: Em caso de empate em uma partida, os envolvidos ganham um ponto. Para o caso em que uma equipe vence, ela irá ganhar 3 pontos enquanto a outra não irá pontuar nada.

Seu programa deve simular um campeonato de futebol. Inicialmente, as equipes devem ser cadastradas. Idealmente, os jogadores devem ser cadastrados e associados às equipes antes das partidas serem iniciadas. Após, as partidas devem ser lidas (mas um jogador pode ser cadastrado a qualquer momento). Seu programa deve armazenar informações das equipes, partidas e jogadores do campeonato. Essas informações serão utilizadas posteriormente.

Equipes são cadastradas quando o programa recebe um caractere “E”, com um identificador único e nome da nova equipe na linha abaixo:

E
1 CCFC
E
2 ECFC

Jogadores são cadastrados quando o programa recebe um caractere “J”, seguido de 3 informações na linha abaixo: identificador único, identificador único da equipe e nome do jogador

```
J
1 1 Fulano
J
2 2 Fulano Júnior
```

Já as partidas são cadastradas quando o programa recebe um caractere “P”, seguido de ao menos uma linha. Na primeira linha: identificador único da partida, identificador único da equipe 1, identificador único da equipe 2, quantidade de gols da equipe 1 e quantidade de gols da equipe 2. Para o caso em que houveram M gols na partida, M linhas devem ser digitadas, onde cada uma traz o id do jogador que fez um gol. Exemplo:

```
P
1 1 2 3 1
1
1
1
2
P
2 2 1 0 0
```

No exemplo acima, estamos lendo a informação de uma partida (id único 1) entre os times CCFC (id único 1) e ECFC (id único 2), onde CCFC fez 3 gols e ECFC fez 1. Os 3 gols de CCFC foram feitos pelo jogador “Fulano” (id único 1) e o gol de ECFC foi feito pelo “Fulano Júnior” (id único 2). Em seguida, lemos a informação de uma outra partida (id único 2), entre os mesmos times, que terminou sem gols (0x0);

Observação:

- Caso um time esteja envolvido em uma partida e tenha menos de 5 jogadores, ele deve perder automaticamente (por W.O). Considere que, se isso acontecer, a partida “não inicia” e portanto, seu placar é 0x0 (não há gols);
- Considere que, caso isso aconteça, apenas um dos times terá menos de 5 jogadores. Essa “regra” faz com que a entrada mostrada nesse roteiro não seja válida (CCFC e ECFC possuem apenas um jogador) como caso de teste. Ela é mostrada aqui apenas para
- Olhe os casos de teste fornecidos!
- Considere que não haverá identificadores únicos repetidos para o mesmo tipo de dado. Ou seja, pode haver um time, um jogador e uma partida com id único “1”, mas nunca haverá dois times com id único “1”.
- Considere também que não há erros nos ids;

- Propositadamente, um jogador não armazena quantos gols fez e nem um time armazena seus jogadores. Esse processamento faz parte das funções dos TADs!

Escreva um programa em C (main.c) que leia os dados de jogadores, equipes e partidas e em seguida mostre qual foi o time campeão (pontuou mais) e o jogador artilheiro (fez mais gols) daquele campeonato, utilizando os TADs fornecidos. Os dados devem ser lidos conforme descrito acima, até que um “F” seja digitado, indicando fim do cadastro.

Ao finalizar, um processamento deve ser feito e uma das seguintes saídas deve aparecer:

- Caso o campeonato esteja incompleto – isto é, não houve cadastro de $N^2 - N$ partidas (N = número de equipes), o programa deve imprimir “CAMPEONATO NAO FINALIZADO”
- Caso esteja completo, mas houve algum erro de digitação – isto é, foram cadastradas todas as partidas, mas há ao menos um time com menos jogos que outros, o programa deve imprimir “ERRO DE CADASTRO” (ou se algum time não tiver $2*(N-1)$ partidas)
- Caso esteja completo e não houve erros de digitação (ou seja, não houve inconsistências) – todos os times possuem o mesmo número de partidas, as seguintes linhas devem ser impressas em tela:

```
EQUIPE CAMPEA: N (#ID) P:# J:# V:# E:# D:# GP:# GC:# SG:# AP:%
ARTILHEIRO: NOME_JOGADOR (NOME_EQUIPE) - # GOLS
```

Onde:

- N – nome da equipe
- ID da equipe
- P – nº pontos
- J – nº partidas jogadas
- V – nº vitórias
- E – nº empates
- D – nº derrotas
- GP – nº gols pró
- GC – nº gols contra
- SG – saldo de gols (diferença entre gols)
- AP – aproveitamento (2 casas decimais)

Ex:

```
EQUIPE CAMPEA: CCEC(1) 4 2 1 1 0 3 0 3
ARTILHEIRO: FULANO (CCEC) - 3 GOLS
```

Dicas:

- Todo scanf finaliza com um \n, independente do tipo.
- Todos os nomes (seja time ou jogador) podem conter espaços. Utilize %[^\n] no scanf;
- Para imprimir o caractere % no printf, utilize %%
- Vocês tem acesso aos casos de teste e *script* de correção. Use-os com inteligência.
- A main consiste na, leitura das entradas, armazenamento e processamento dos dados, verificação de inconsistências e, caso não haja nenhuma, geração do relatório final. Para as duas últimas etapas, há bibliotecas específica com as funções; As demais devem (leitura, armazenamento e processamento) ser feitas na própria main.
- A leitura é feita através de uma função específica para cada TAD, e o armazenamento consiste em salvar o retorno dessa função em um vetor, por exemplo.
- Já o processamento consiste em armazenar os “efeitos colaterais” que um novo cadastro causa em outro TAD. Por exemplo, armazenar que um time tem um jogador adicional após o cadastro desse jogador, ou alterar os campos de uma equipe para registrar vitórias, derrotas, empates, autores dos gols, etc;
- Considere que não há caracteres especiais (acentos, ç, etc) na entrada/saída.

Regras gerais

- A atividade é **individual**. Todas as questões serão testadas e plágio não será tolerado;
- Equipamentos eletrônicos pessoais devem estar guardados na mochila.
- Seu programa deve, obrigatoriamente, utilizar o *template* disponibilizado junto com o exercício. Isso significa que você não deve alterar os arquivos .h;
- A correção é feita majoritariamente utilizando o *script* de correção
- Se programa deve simular o encapsulamento dos TADs. Caso sejam detectados acessos diretos aos campos das estruturas, haverá desconto na nota.
- Todos os valores máximos de arrays são fornecidos nos *templates*