



Q1	2,0	
Q2	2,0	
Q3	2,0	
Q4	2,0	
Q5	2,0	

Fundação CECIERJ – Vice Presidência de Educação Superior à Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação de Aplicações Web

Professores: Flávio L. Seixas e Miguel Elias M. Campista

APX2 – 1º Semestre de 2022

Orientações para a prova:

1. **Mostre todos os passos para se chegar a solução encontrada.**
2. **Você pode: (A) escrever de forma legível as suas respostas em folhas em branco com caneta preta, escaneá-las e convertê-las para PDF, ou (B) escrever as suas respostas no computador usando um editor de texto, e converter o arquivo para PDF.**
3. **Procure numerar todas as folhas e indicar a questão a que a resposta se refere.**
4. **As respostas serão individuais. Ou seja, não é permitido consultas ao tutor, material da disciplina, websites, ou outra pessoa.**
5. **O arquivo PDF deve ser anexado junto à entrega da tarefa APX2.**
6. **O prazo de entrega da tarefa é dia 28/mai, até às 12h (horário de Brasília).**

Esta AP avalia o uso das estruturas básicas de repetição e condição, a criação e uso de funções, manipulação de vetores e programação orientada a objetos, modelagem de banco de dados, consultas, inserções, remoções e atualizações usando SQL, funções PHP para acesso ao banco de dados, validação dos dados e construção de aplicação.

Você está interessado em organizar os jogos de um campeonato de futebol através da implementação de um programa em PHP. Sabe-se que as partidas da fase de classificação já foram registradas em um arquivo CSV, que possui o formato abaixo:

```
<nome do Time X, número de gols do Time X, nome do Time Y, número de gols do  
time Y, grupo dos Times X e Y>
```

O arquivo abaixo é um exemplo do CSV para um campeonato com quatro grupos e três times por grupo. Note que o número de grupos e o número de times por grupo pode variar, mas não o formato de cada registro individualmente (cada linha do arquivo CSV). Considerando que um arquivo CSV com o mesmo formato do abaixo é dado, implemente as funções em PHP, como se pede. **Pode-se escolher entre responder separadamente ou não as Questões Q1 a Q3. Procure deixar claro nas respostas qual questão está sendo respondida.**

```
time A,0,time B,0,grupo 1
time A,1,time C,0,grupo 1
time C,2,time B,0,grupo 1
time D,1,time E,0,grupo 2
time E,1,time F,0,grupo 2
time F,2,time D,3,grupo 2
time G,0,time I,0,grupo 3
time G,1,time H,3,grupo 3
time H,2,time I,0,grupo 3
time J,1,time K,0,grupo 4
time J,1,time L,1,grupo 4
time K,2,time L,3,grupo 4
```

Q1. Implemente uma função em PHP que seja capaz de ler o arquivo CSV dado e organizar as partidas em uma estrutura de array associativo como a mostrada abaixo.

```
$tabelaGrupos = Array (
    [grupo 1] => Array (
        [0] => Array ( [0] => time A [1] => 0 [2] => time B [3] => 0 )
        [1] => Array ( [0] => time A [1] => 1 [2] => time C [3] => 0 )
        [2] => Array ( [0] => time C [1] => 2 [2] => time B [3] => 0 )
    )
    [grupo 2] => Array (
        [0] => Array ( [0] => time D [1] => 1 [2] => time E [3] => 0 )
        [1] => Array ( [0] => time E [1] => 1 [2] => time F [3] => 0 )
        [2] => Array ( [0] => time F [1] => 2 [2] => time D [3] => 3 )
    )
    [grupo 3] => Array (
        [0] => Array ( [0] => time G [1] => 0 [2] => time I [3] => 0 )
        [1] => Array ( [0] => time G [1] => 1 [2] => time H [3] => 3 )
        [2] => Array ( [0] => time H [1] => 2 [2] => time I [3] => 0 )
    )
    [grupo 4] => Array (
        [0] => Array ( [0] => time J [1] => 1 [2] => time K [3] => 0 )
        [1] => Array ( [0] => time J [1] => 1 [2] => time L [3] => 1 )
        [2] => Array ( [0] => time K [1] => 2 [2] => time L [3] => 3 )
    )
)
```

Note que o array associativo utiliza o nome do grupo como chave e um array de arrays como valor. Abaixo está uma chamada à função `divideGrupos`. A função

recebe o nome do arquivo CSV e o índice da coluna referente ao nome do grupo como argumentos para retornar a estrutura solicitada.

```
$tabelaGrupos = divideGrupos ("equipes.csv", 4);
```

Q2. Assumindo que os arquivos CSV possam ser corrompidos, implemente uma função em PHP que verifique se todos os times jogaram o mesmo número de partidas na fase de classificação do campeonato. Essa função deve receber como argumento a estrutura calculada na **Questão Q1** e os índices das colunas referentes aos nomes dos times como argumentos. A função retorna verdadeiro caso o número de partidas seja igual para todos os times e falso, caso contrário. No exemplo, todos os times jogam duas partidas. Abaixo está um exemplo de chamada à função `verificaTimes`:

```
$resultadoBooleano = verificaTimes ($tabelaGrupos, 0, 2);
```

Q3. Implemente uma função `computaMelhoresColocados` que retorna a estrutura abaixo:

```
$tabelaGruposOrdenados = Array (
    [grupo 1] => Array (
        [time A] => 4 [time C] => 3 [time B] => 1
    )
    [grupo 2] => Array (
        [time D] => 6 [time E] => 3 [time F] => 0
    )
    [grupo 3] => Array (
        [time H] => 6 [time G] => 1 [time I] => 1
    )
    [grupo 4] => Array (
        [time J] => 4 [time L] => 4 [time K] => 0
    )
)
```

Note que a estrutura é do tipo array associativo que usa o nome do grupo como chave e o valor é um array associativo dos times ordenados conforme a pontuação obtida. Cada vitória vale 3 pontos e cada empate vale 1 ponto. As equipes derrotadas não somam pontos. Uma dica é usar a função `arsort` do PHP para ordenar arrays associativos conforme o valor.

Observe o protótipo da função `computaMelhoresColocados` que recebe como argumentos a estrutura da **Questão Q1** e os índices das colunas referentes aos nomes dos times. Assume-se que o índice da coluna com o número de gols em cada partida é igual ao índice do nome do time + 1 e que a função `computaMelhoresColocados` verifica se os times possuem o mesmo número de partidas (**Questão Q2**). Caso não possua o mesmo número de partidas, um array vazio deve ser retornado. Em caso de empate no número de pontos entre dois times do mesmo grupo, qualquer critério de ordenação pode ser usado.

```
$tabelaGruposOrdenados = computaMelhoresColocados($tabelaGrupos, 0, 2);
```

Assuma agora que os dados do arquivo CSV foram carregados em um banco MySQL e que duas tabelas foram implementadas: uma para Times e outra para Partidas. A Tabela Times possui um identificador primário, o nome da equipe, o nome do grupo, o número de vitórias, empates e derrotas. Já a Tabela Partidas tem os identificadores dos dois times participantes, o número de gols de cada uma das equipes na partida. Dessa forma, programe as seguintes consultas:

Q4. Implemente em MySQL uma consulta que retorne o nome do time e o número de pontos correspondente do melhor time na fase de classificação do campeonato (considerando todos os grupos). O melhor time é aquele que conquistou o maior número de pontos. Caso haja empate, a consulta deve retornar o nome e o número de pontos de todos os times empatados.

Q5. Implemente em MySQL uma consulta que retorne o nome do time e o saldo de gols dos melhores colocados de cada um dos grupos.