
Avaliação 2 (av_3)

1. Crie uma estrutura *Time* com os seguintes campos: **int** *codigo*, **char** *nome*[50], **int** *partidas*[10], **int** *gols_feitos* e **int** *gols_sofridos* . Os elementos do vetor *partidas* devem ter apenas os valores 1 (vitória), 0 (empate) e -1 (derrota). Levando em consideração essa estrutura:
 - (a) Crie uma função que tem um vetor de times e o seu tamanho parâmetros. A função deve retornar o código do time com a pior defesa. Ou seja, o código do time que sofreu mais gols. (1,0)
 - (b) Crie uma função que tem como parâmetro um time e retorna a quantidade de pontos que ele possui. Vale salientar que um time tem 3 pontos por vitória, 1 ponto por empate e 0 pontos por derrota. (1,0)
 - (c) Crie uma função que tem um vetor de times como parâmetro e o seu tamanho. A função deve retornar o código do time que tem mais pontos. (1,5).
 - (d) Crie uma função que tem um vetor de times como parâmetro e o seu tamanho. A função deve retornar a quantidade de times que tem saldo de gol positivo. (SaldoGols = golsFeitos - golsSofridos) (0,5)
 - (e) Crie uma função que tem um vetor de times como parâmetro e o seu tamanho. Essa função deve alocar um vetor de inteiros na memória heap que contém os códigos dos times que possuem saldo de gols positivo. A função deve retornar o endereço do vetor alocado na memória heap.(2,0)
 - (f) Crie um procedimento que tem um vetor de times como parâmetro e o seu tamanho. Ele deve imprimir os times que ganharam pelo menos 3 partidas seguidas. Por exemplo, o time "Juazeiro do Norte", com o vetor de partidas [1, 0, 0, **1**, **1**, **1**, 0, -1, 1, 0] deve ser impresso (2,0)
2. Faça um algoritmo que leia uma matriz de inteiros $n \times n$. O algoritmo deve mostrar a soma dos elementos que estão acima da matriz principal da matriz. (2,0)