

Lista de Exercícios - Revisão Lógica de Programação

Prof. Thiago Cavali

27/07/2023

1. Elabore um programa que a partir das pontuações do líder e do lanterna de um campeonato de futebol mostre quantas **vitórias** são necessárias para que o lanterna **iguale ou ultrapasse** o líder na classificação.
Observações:
 - Cada vitória conta 3 pontos;
 - O usuário deve fornecer as pontuações;
 - O usuário deve ser notificado do erro caso a pontuação do líder seja inferior à do lanterna;
 - No caso anterior, deve-se solicitar que o usuário insira novos valores.
2. A partir do exercício anterior, determine o número de **vitórias e empates** necessários para o lanterna **igualar** a pontuação do líder. Observação: cada empate conta 1 ponto.
3. Em determinada disciplina a avaliação é feita com duas provas e um projeto, todos avaliados de 0 a 10 pontos. Para o cálculo da média final (MF) a média aritmética simples das duas provas tem peso 7 e a nota do projeto peso 3. Os estudantes serão considerados aprovados caso MF seja maior ou igual a 6 pontos e reprovados caso contrário. Com base no descrito, escreva um programa que receba do usuário o nome do aluno e as notas de suas 3 avaliações e retorne uma saída no formato: "Nome - MF - Situação" (onde situação é "Aprovado" ou "Reprovado").
4. Para se aposentar em 2023 as mulheres precisam ter na soma Idade + Tempo de Contribuição um total de 89 anos, além disso o Tempo de Contribuição deve ser no mínimo de 30 anos. Já para os homens a soma deve ser de 99 anos com no mínimo 35 anos de contribuição. Sabendo disso, elabore um programa que receba do usuário seu sexo, idade e tempo de contribuição e retorne se o usuário em questão pode ou não se aposentar em 2023;
5. Elabore um programa que simule um sistema de acesso com login e senha. O usuário deve informar seu login e em seguida tem 3 tentativas para acertar a senha padrão (Considere a senha padrão "senhapoucosegura"). Após a inserção da senha considere:
 - Caso o usuário acerte a senha, uma mensagem de boas vindas (contendo o nome do usuário) deve ser exibida;
 - Caso erre pela primeira ou segunda vez, deverá ser mostrada uma mensagem informando o erro e o número de tentativas de acesso restantes;
 - Caso erre pela terceira vez, a mensagem mostrada deverá informar o bloqueio daquele usuário por excesso de tentativas de acesso.
6. Elabore um programa que receba um número do usuário e retorne a tabuada daquele número.
7. Elabore um programa que simule um jogo simples de dados de 6 faces entre dois jogadores. Nesse jogo os jogadores A e B lançam os dados e quem obtiver o maior valor ganha. Caso o número seja o mesmo para ambos os jogadores ocorre um empate. A saída da classe deve ser uma mensagem mostrando qual número cada jogador obteve e anunciando o resultado (vitória de A, B ou empate). Observações:

- Utilizar um gerador de números aleatórios;
 - Não é necessária a entrada de dados do usuário.
8. Construa uma matriz que represente uma imagem de 10x10 pixels. Cada pixel deve conter um valor (gerado aleatoriamente) entre 0 e 255 representando o tom da cor presente naquele pixel. Exibir essa matriz na tela no formato de tabela.
9. Faça um programa que faça o controle de um estacionamento com capacidade para 10 carros. Essa classe deve ser controlada por um menu (rodando em loop) que permita ao usuário:
- Adicionar um novo carro. Nesse caso o usuário deve fornecer o número da vaga e a placa do carro. Caso a vaga esteja livre o carro deverá ser colocado na vaga, caso contrário, o usuário deverá ser informado;
 - Liberar um novo carro. O usuário deverá fornecer um número de vaga e a placa daquela vaga deverá ser retirada;
 - Verificar status. Deve retornar a ocupação atual do estacionamento no formato "número da vaga - Placa do carro" para as vagas ocupadas ou "número da vaga - Vazio" para as vagas vazias;
 - Sair do programa.

Considerações Gerais:

- Ao terminar, chamar o Professor para conferir a resolução e explicar o desenvolvimento;
- A entrega deverá ser feita via AVA na Tarefa "Lista de Exercícios - Revisão Lógica de Programação" em um único arquivo compactado (.rar/.zip/.7z/.tar) com nome no formato "Nome - RA";
- Esse arquivo deve conter os arquivos dos projetos.