TRABALHO
Linguagens de Programação
Prof. Lucas Ismaily

ORIENTAÇÃO A OBJETOS (10 pontos)

1. Atualmente, o pessoal da UFC mantém um grupo de futebol, o nível não é dos melhores, a disposição física tão pouco, em compensação a criação de memes é de altíssima qualidade. Sua tarefa é implementar um sistema que gerencia a formação dos times. Cada jogador possui nome, apelido e uma pontuação que reflete a sua destreza. O nível tem uma escala de 1 a 5 (sendo 1 o nível canelada, e 5 o nível doutrinador do racha). O sistema deve permitir o cadastro de jogadores e a atribuição dos níveis. Além disso, o sistema deve diferenciar jogador linha e goleiro. O sistema também contém uma entidade time, que contém um número, um conjunto com jogadores e um goleiro. A principal funcionalidade é montar um racha, que é composto por uma data, local, alocação dos times e um placar do racha (quantas vitórias cada time obteve). Para alocar os times, o sistema recebe o número de jogadores de linha, e deve montar times balanceados, isto é, os níveis dos times são equilibrados, sendo que o nível de um time é dado pela soma dos níveis dos atletas.

   Não precisa utilizar banco de dados. Também não precisa criar uma interface gráfica. Contudo, é necessário uma navegabilidade no sistema mesmo que via teclado.

LINGUAGEM FUNCIONAL (10 pontos)

Você pode utilizar Python (de modo funcional) ou qualquer outra linguagem funcional para resolver as seguintes questões. Importante: é **expressamente proibido** o uso de instruções de repetição e atribuições de variáveis para usos futuros (variáveis auxiliares). Em resumo, o seu código deve ser o mais funcional possível. Ademais, o uso da instrução *if/else* é permitido.

1. Faça um programa que receba duas listas A e B e imprima a diferença simétrica de A e B. A *diferença simétrica* de A e B é dada por $(A\text{-}B)$ U $(B\text{-}A)$.

2. Faça um programa que receba uma lista de inteiros L e dois números p e q inteiros e imprima a lista L sendo que toda ocorrência de p deve ser trocada por $p+q$.

3. Faça um programa que recebe um inteiro positivo n e uma *string* W e determina se W contém uma *substring* de tamanho maior ou igual a n que é palíndroma. A saída deve

ser apenas as palavras *sim* ou *nao*.

4. Faça um programa que receba uma lista de inteiros e imprima somente os números que são primos. Um número inteiro maior que um é *primo* se contém exatamente dois divisores inteiros positivos.

5. Faça um programa que receba como entrada coordenadas de um retângulo no plano e um ponto *p* no plano, e imprima 'sim' se o ponto *p* está dentro do retângulo e 'nao' caso contrário. Suponha que o retângulo seja representado por uma tupla (int, int, int, int), contendo respectivamente as coordenadas *x* e *y* do ponto no seu canto inferior esquerdo, seguidas das suas medidas de largura e altura. Considere, ainda, que o eixo *y* cresce de baixo para cima e o eixo *x* cresce da esquerda para direita.

OBS.: A entrada de todas as questões é pelo teclado (entrada padrão), e é digitado apenas a entrada, sem nenhum caractere especial. Por exemplo, se a entrada for uma lista numérica, deverá ser digitado 1 2 3 5 4 e não [1, 2, 3, 5, 4], isto é, apenas os números separados por espaço são digitados. Outrossim, o único valor impresso na tela deve ser o resultado. Qualquer impressão diferente do resultado, como por exemplo, "digite a entrada" ou "a saída é" será considerado **resposta errada**. Novamente, imprima apenas o resultado. Por exemplo, na Questão 6, se a entrada for 1 2 11 6 5, a única impressão na tela será 2 11 5. Caso a resposta seja vazia, você deve imprimir apenas um espaço em branco. Por exemplo, se na entrada não houverem números primos, sua impressão será apenas um espaço em branco. Ademais, nas respostas que são listas, a única coisa impressa são os elementos de forma horizontal e separados por espaços, por exemplo, 1 2 3 4 em vez de [1, 2, 3, 4]. Novamente e repetidamente, qualquer impressão ou entrada de dados que não seguir o padrão explicado aqui, será considerado **resposta errada.** Se está em dúvida quanto a saída e/ou entrada dos dados, pergunte.

SOBRE PROGRAMAÇÃO LÓGICA (10 pontos)

1. Utilizando Prolog implemente um predicado *primos*(*L*) que determina se todos os números na lista *L* são primos.

2. Utilizando Prolog implemente um predicado *n_esimo*(*N*, *L*, *X*), onde *X* é o *N*-ésimo elemento da lista *L*.

3. Utilizando Prolog implemente o predicado *tamanho*(*L*, *X*), onde *X* é comprimento da lista *L*.

4. Implemente o predicado *elimina_repetidos*(L_1 , L_2) que recebe em L_2 o resultado de L_1 sem elementos duplicados. Por exemplo, se L_1 = [1 , 1 , 1 , 5 , 5], então L_2 = [1 , 5].

5. Implemente um programa em Prolog sobre a seguinte família:

   Construa um banco de fatos de uma família. De modo que seja possível representar os seguintes predicados: sexo(masculino ou feminino), irmã, irmão, descendente, mãe, pai, avô, avó, tio, tia, primo e prima.

INFORMAÇÕES IMPORTANTES

A nota do trabalho será a média aritmética das notas nos três paradigmas: Orientação a Objetos, Funcional e Lógico. Além disso, será selecionado um trabalho de OO para ser implantado para gerenciar os rachas da UFC. O aluno cujo sistema foi selecionado terá 2 pontos acrescentados à nota final do trabalho. O futebol apresentado pode ser de qualidade duvidosa, mas a tecnologia implantada é de primeira (ou não, vai depender da turma).

A entrega será **somente** pelo SIPPA, numa pasta zipada com todas as questões (no caso da questão de Orientação a Objetos, você pode zipar o projeto). **Importante:** cada questão de funcional e lógico deve está em um arquivo separado. O prazo máximo da entrega do trabalho é para o dia **06/07/2023.**

Trabalho individual. Sejam honestos com vocês e comigo, por favor. Se for detectado qualquer tipo de fraude, os envolvidos receberão nota **zero. Note: os envolvidos receberão zero, não importa se você foi a origem ou o destino, ambos receberão nota zero.**