



116343 Linguagens de Programação – Turma A – Semestre 2014/2

Prova Prática de SWI Prolog

A prova é individual e qualquer cola ensejará nota zero, inclusive cópia da Internet. Responda apenas ao número de questões solicitado. As respostas devem ser enviadas em um único arquivo (com extensão pl), cujo nome deve ser a matrícula do aluno, até as 17h30min, via thread do Moodle Aprender. Não serão aceitas submissões após o fechamento da thread.

I Parte – Resolva apenas a DUAS questões (2,0)

1. Crie um programa que receba uma lista de 4 elementos e verifique se todos os elementos são iguais. (0,5)
2. Crie um programa que retorne a lista de todos os divisores de um dado número. (1,0)
3. Crie um programa que verifique se duas listas possuem os mesmos elementos (a ordem não importa). (0,5)
4. Crie um programa que retorne o número de vezes que um dado elemento aparece numa lista. (1,0)
5. Crie um programa que transforme uma lista em palíndromo. Ex: transforma [1,2]. ?- [1,2,2,1] (0,5)
6. Crie um programa que receba uma lista e retorne o número de elementos acima da média deles. (1,0)

II Parte - Questões obrigatórias

7. Com base nos seus familiares construa uma base de conhecimento que contenha apenas os predicados genitor/2, homem/1 e mulher/1, contendo no mínimo 50 pessoas. Expanda essa base com a criação do máximo de relações familiares que imaginar. Por exemplo, a relação mãe pode ser construída com o predicado mae/2 como: (5,0)

mae(X,Y) :- mulher(X), genitor(X,Y).

Critérios de correção: construção da base (1,0), corretude (3,0), documentação interna (0,5), flexibilidade e confiabilidade do código (0,5). A nota da corretude dependerá do número de relações que forem criadas. O mínimo que deverá ser criado são 10 (dez) relações, garantindo assim 2,0 pontos na nota de corretude. Caso sejam criadas mais relações, será dado um adicional de 0,2 por relação, podendo chegar até a 3 pontos. Caso apresente o número de ancestrais, para cada pessoa, receberá mais 1,0 ponto.

8. Crie um programa que requer o uso do operador “cut” (1,0) e justifique porque ele é necessário (1,0). (2,0)
9. Crie um programa que utilize o laço “repeat-fail” e cláusulas disjuntivas (operador “ou”). (2,0)
10. Crie um programa que verifique se um ponto (x,y) está dentro de uma circunferência de raio R e com centro localizado num ponto (a,b). (1,0)

Boa prova!