

UNIVERSIDADE FEDERAL DE SÃO CARLOS - DEPARTAMENTO DE COMPUTAÇÃO
PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO
Programação Lógica e Funcional - Primeiro Trabalho/2009

PARTE 1 – Escreva um programa PROLOG com os seguintes procedimentos:

1. Definir uma cláusula *reajustatabela* (*Tabela*, *Porcentagem*, *Produtos*, *NovaTabela*), para atualizar uma tabela de preços alterando o preço de alguns produtos dessa tabela, de acordo com uma porcentagem, sendo que:
 - *Tabela* é uma lista de pares de elementos (também listas) onde o primeiro elemento do par é um produto e o segundo é um preço;
 - *Porcentagem* é um número que indica qual o acréscimo do preço sendo reajustado;
 - *Produtos* é uma lista de produtos que devem ter seu preço reajustado;
 - *NovaTabela* é a tabela resultante depois dos reajustes.

```
?- reajustatabela([[cafe, 1.50], [suco, 1.80], [vitamina, 2.20], [esfirra, 1.50]], 0.2,
[esfirra, vitamina], Novatabela).
Novatabela = ([[café, 1.50], [suco, 1.80], [vitamina, 2.64], [esfirra, 1.80]])
```

A lista de produtos que representa a tabela pode ser inicializada por fatos e os valores de porcentagem e produtos a serem reajustados podem ser dados via teclado.

2. Definir uma cláusula *shift*(*N*, *Lista*, *ListaNova*) para fazer a rotação dos elementos de uma lista dada para a direita, um certo número de posições, sendo que:
 - *D* indica se a rotação deve ser feita para a esquerda ou para a direita;
 - *N* indica quantas posições a lista deve rodar;
 - *Lista* é a lista dada, que pode ter qualquer tipo de elementos, inclusive outras listas;
 - *ListaNova* é a lista resultante.

Para os exemplos abaixo, o resultado seria:

```
?- shift(3, [a, b, 86, [c,d], xyz], LN).
```

```
LN = [86, [c,d], xyz, a, b]
```

```
?- shift(8, [ b, 4, [a, b, c] , [ ], 34.7, 1000], LN).
```

```
LN = [34.7, 1000, b, 4, [a, b, c] , [ ]]
```

Os predicados de entrada e saída podem ser usados para ler do teclado a lista dada e o número de posições. As listas dadas podem opcionalmente ser inicializadas como fatos e recuperadas a partir de um nome dado via teclado.

PARTE 2 - Escreva um programa LISP composto das seguintes funções:

1. Função que testa se duas listas (possivelmente com sublistas) dadas tem a mesma estrutura. Duas listas que tem a mesma estrutura podem ter átomos diferentes. Se um dos dois argumentos dados não for lista, a função deve retornar nil.

Exemplos:

Lista 1	Lista 2	Estruturas iguais?
()	()	sim
()	(a b c)	não
(a b (c d (e f g) (h ((i) j) l ())))	(x n l (dia d (e l go) (h ((inter) z) 2 ())))	sim
(a f () (g l))	((a f) () g l)	não

2. Função com dois parâmetros, um átomo e uma lista, que retorna o número de vezes que o átomo aparece na lista, em todos os níveis.

Observações:

- Os trabalhos podem ser feitos em duplas, que deverão ser as mesmas em todos os trabalhos;
- Usar OBRIGATORIAMENTE:
 - LPA Prolog para windows ou SWI-PROLOG para o trabalho de Prolog e
 - CLisp ou LISP works para o trabalho de LISP; Não serão aceitos trabalhos em outro software;
- Entregar o código fonte e resultados de execução de pelo menos dois exemplos para cada item do trabalho via Moodle, em arquivo texto.
- Data de entrega:20/05/2009.