MC336 B e MC600 - Prova Lisp e Prolog - 1s2005

1) Lisp (2,5 pontos). Dado uma lista escreva a função `embaralha` que embaralha lista (veja exemplos abaixo)

```
(embaralha '(1 2 3 4 5 6))        <- numero par de elementos
=> (1 4 2 5 3 6)

(embaralha '(1 2 3 a b))          <- numero impar de elementos
=> (1 a 2 b 3)

(embaralha '())
=> ()
```

2) Prolog (2,5 pontos). Escreva o predicado `embaralha(L,E)` que dado uma lista como primeiro argumento, será verdadeiro se a segundo argumento for o resultado de embaralhar a lista. O predicado deve funcionar, pelo menos, no modo +-

```
embaralha([1,2,3,4,5,6],X)
X = [1,4,2,5,3,6]

embaralha([1,2,3,a,b],X)
X = [1,a,2,b,3]
```

3) Lisp. (2,5 pontos) Dado uma lista que contem sub-listas com 2 elementos, onde o primeiro elemento é o nome do produto (um símbolo) e o 2 a quantidade do produto achado no deposito. O mesmo produto pode aparecer um várias sub-listas, com quantidades diferentes, pois ele pode estar em vários cantos do deposito. Escreva a função `produtos` que retorna a lista dos produtos e suas quantidades totais em ordem decrescente de quantidade.

```
(produtos '((a 3) (a 4) (b 5) (a 7) (c 1) (c 1)))
=> ((a 14) (b 5) (c 2))
```

4) Prolog. (2,5 pontos) Dado uma lista que contem sub-listas com 2 elementos, onde o primeiro elemento é o nome do produto (um símbolo) e o 2 a quantidade do produto achado no deposito. O mesmo produto pode aparecer um várias sub-listas, com quantidades diferentes, pois ele pode estar em vários cantos do deposito. Escreva o predicado `produtos(Listas,Resumo)` que é verdadeiro quando o 1o argumento é uma lista como a descrita acima e a segunda uma lista dos produtos e suas quantidades totais em ordem decrescente de quantidade (se não for em ordem decrescente a questão valerá no maximo 2 pontos). Modo +-

```
produtos([[a, 3], [a, 4], [b, 5], [a, 7], [c, 1], [c, 1]],L)
L =  [[a, 14], [b, 5], [c, 2]]
```