Paradigmas da Programação III
Exame 1^a chamada 1997/98
LESI e LMCC 2^o Ano

Responda às seguintes questões:

1. Considere o seguinte conjunto de cláusulas:

   $member(X, [X|R])$

   $member(X, [H|T]) \Leftarrow member(X, T)$

   a) Converta cada uma destas fórmulas na forma disjuntiva.

   b) Usando a **resolução**, demonstre que deste conjunto de cláusulas se infere $member(X, [a, b])$, indicando explicitamente as unificações necessárias.

2. Considere o seguinte programa Prolog:

```
ap([], X, X).
ap([X|R], Y, [X|T]) :- ap(R, Y, T).

ap1([], X, X) :- !.
ap1([X|R], Y, [X|T]) :- ap1(R, Y, T).
```

   Desenhe as árvores de procura SLD para as seguintes questões:

   a) `ap(X, Y, [a,b]).`

   b) `ap1(X, Y, [a,b]).`

3. É frequente os livros incluirem um índice remissivo onde é feita a associação entre uma palavra e os números das páginas onde essa palavra ocorre. Admitindo que esta informação está armazenada numa lista de pares *(Palavra, Lista_de_num_de_paginas)*, defina:

   a) Um predicado *acrescenta(Indice, Palavra, Pagina, Indice1)* que sucede se *Indice1* é o resultado de acrescentar uma associação *Palavra-Pagina* ao *Indice*.

   b) Um predicado *pags(Indice, Palavra, ListaPags)* que sucede se *ListaPags* é a lista de páginas associada a *Palavra*, no *Indice*.

   c) Um predicado *pals(Indice, Pagina, Palavra)* que, por backtracking, indique as palavras do *Indice* referenciadas numa dada *Pagina*.

4. Defina um predicado Prolog *in_order(Arvore, Lista)* onde *Lista* é o resultado de fazer uma travessia *in_order* da árvore binária *Arvore*. Por exemplo:
   $in\_order(tree(a, tree(b, void, void), tree(c, tree(d, void, void), tree(e, void, void))), [b, a, d, c, e])$.

5. Dado um mapa de ligações directas entre cidades, sob a forma de um conjunto de factos Prolog da forma *ligacao(Cidade1, Cidade2)*, escreva um programa Prolog para determinar (por backtracking) os caminhos directos ou não entre duas cidades. O programa deve evitar os ciclos existentes no mapa.