

---

---

## PARADIGMAS DE PROGRAMAÇÃO III

2003/04 - Janeiro (1ª chamada)

LESI - 2º Ano

17 de Janeiro de 2004

Duração Máxima: 2 horas

---

---

Obs: Responda a cada uma das partes em folhas separadas e bem identificadas

### PARTE I

#### 1: LISTAS e RECURSIVIDADE. (6 pontos)

1.1.- Dada uma lista L contendo listas de inteiros, escreva:

- a) Um predicado *maiorLista/2* que dê como resultado a maior lista contida na lista parâmetro.
- b) Um predicado *posMenor/2* que dê como resultado a posição da lista de menor comprimento da lista parâmetro (head = posição 1).

1.2.- Dada uma lista L contendo inteiros e 2 inteiros X e Y, escreva um predicado *segmentos/6* que dê como resultado 3 listas, contendo, respectivamente, todos os elementos de L menores que X, entre X e Y inclusivé, e maiores que Y.

#### 2: Semântica do Prolog. (1.5 pontos)

A semântica lógica de um dado programa traduz-se pela expressão:

$$P \Leftrightarrow (A \wedge B \wedge C) \vee (\sim A \wedge D \wedge E) \vee (\sim D \wedge F \wedge G)$$

Escreva um programa Prolog com correspondente semântica.

### PARTE II

#### 3: Queries e Múltiplas Soluções. (7 pontos)

Numa Base de Conhecimento são registados factos respeitantes a *alunos* e *disciplinas*, respectivamente sob a forma *aluno(num, nome, curso/ano/inscricao)*, em que inscrição é uma lista de códigos de disciplinas, e *disciplina(cod, curso, ano)*.

Escreva os predicados necessários à satisfação dos seguintes "queries" genéricos e seguintes transformações da BC:

- a) *Quantos alunos frequentam a disciplina de código D ?*
- b) *Qual a disciplina do ano A com mais alunos inscritos ?*
- c) *Lista de pares Ano/Nº Total de alunos para um dado Curso ?*
- d) *Eliminar da BC todas as disciplinas cujos códigos figuram numa lista parâmetro L.*
- e) *Ler de um ficheiro novas inscrições de alunos, acrescentando os novos à BC (novo código) e actualizando as informações dos já existentes (código existente).*

### PARTE III

#### 5: Grafos. (5.5 pontos)

Numa BC foram introduzidos predicados **instalacao(codsw, lista\_codsw)**, onde *codsw* representa um código de um produto software e *lista\_codsw* é uma lista de códigos de produtos. Cada facto *instalacao/2* representa todos os produtos de software necessários para instalar o software cujo código é o 1º parâmetro. Um produto independente de outros é representado por **instalacao(codsw, [])**, ou seja, a sua instalação não depende de qualquer outro.

- a) A partir de uma BC contendo apenas factos *instalacao/2*, crie um grafo de dependências entre produtos software, no qual cada nodo contém o código de um dado produto de software e cada arco orientado (que liga este nodo a outro) representa a dependência entre este produto e o do nodo destino. Cada ligação deste grafo deverá ser representada como **depende(codsw1, codsw2)**.
- b) Considere que para a instalação completa de uma dada aplicação necessita de uma lista L de produtos software. Usando o grafo anteriormente criado (ou em alternativa de último caso os factos *instalacao/2*), determine a lista de todos os produtos independentes que deverá instalar em primeiro lugar.