

# Ficha 7

## Programação Imperativa

### Listas Ligadas

1. Considere a seguinte definição de um tipo para representar listas ligadas de inteiros.

```
typedef struct slist *LInt;  
  
typedef struct slist {  
    int valor;  
    LInt prox;  
} Nodo;
```

- (a) Apresente uma sequência de instruções que coloque na variável **a** do tipo **LInt**, uma lista com 3 elementos: 10, 5 e 15 (por esta ordem).  
(b) Apresente definições (preferencialmente não recursivas) das seguintes funções sobre listas ligadas:
  - i. **LInt cons (LInt l, int x)** que acrescenta um elemento no início da lista.
  - ii. **LInt tail (LInt l)** que remove o primeiro elemento de uma lista não vazia (libertando o correspondente espaço).
  - iii. **LInt init (LInt l)** que remove o último elemento de uma lista não vazia (libertando o correspondente espaço).
  - iv. **LInt snoc (LInt l, int x)** que acrescenta um elemento no fim da lista.
  - v. **LInt concat (LInt a, LInt b)** que acrescenta a lista **b** a **a**, retornando o início da lista resultante).
2. Para gerir a informação sobre os alunos inscritos a uma dada disciplina, é necessário armazenar os seguintes dados:
  - Nome do aluno (string com no máximo 60 caracteres)
  - Número do aluno
  - Nota
  - (a) Defina os tipos **Aluno** e **Turma**. Para o efeito considere que a informação referente aos alunos de uma turma é armazenada numa lista ligada de alunos.
  - (b) Defina uma função **int acrescentaAluno (Turma \*t, Aluno a)** que acrescenta a informação de um dado aluno a uma turma. A função deverá retornar 0 se a operação for feita com sucesso.
  - (c) Defina uma função **Aluno \*procura (Turma t, int numero)** que procura o aluno com um dado número na turma. A função deve retornar **NULL** se a informação desse aluno não existir; caso exista deve retornar o endereço onde essa informação se encontra.
  - (d) Defina uma função que determine quantos alunos obtiveram aproveitamento à disciplina (nota final maior ou igual a 10).