



## Lista de Exercícios sobre Listas Implementadas por Arranjos

- 1) Considere listas implementadas por arranjos, então pede-se para implementar funções que:
  - a. Localize/ Pesquise/Encontre (*search*) elementos
  - b. Concatenar/intercalar (*Merge*) duas listas
  - c. Dividir uma lista em várias (*k*)
  - d. Copiar uma lista
  - e. Ordenar (*sort*) uma lista por ordem crescente/decrescente
- 2) Escreva uma função em C para trocar os elementos *m* e *n* de uma lista (*m* e *n* podem ser chaves ou mesmo ponteiros para os elementos – a escolha é sua).
- 3) Escreva uma função em C para trocar os elementos de índice *m* e *n* de uma lista.
- 4) Escreva uma rotina, *inssub(l1,i1,l2,i2,len)* para inserir os elementos da lista *l2*, começando no elemento *i2* e continuando por *len* elementos na lista *l1*, começando na posição *i1*. Nenhum elemento da lista *l1* deverá ser removido ou substituído. Se  $i1 \geq \text{length}(l1)$  (onde  $\text{length}(l1)$  indica o número de nós na lista), ou se  $i2 + len > \text{length}(l2)$ , ou se  $i1 < 0$ , ou se  $i2 < 0$ , ou se  $i1 + len > \text{maxlength}(l1)$  (onde  $\text{maxlength}(l1)$  indica o número máximo de nós na lista *l1*), imprima uma mensagem de erro. A lista *l2* deve permanecer inalterada.
- 5) Você foi contratado pela *Google* para refazer o Orkut. A sua primeira tarefa é implementar uma estrutura de dados para representar as relações de amizade dentro do Orkut. Ou seja, para cada pessoa cadastrada você tem que armazenar a sua lista de amigos. Considere que cada pessoa no Orkut tem um identificador único que é um número inteiro.
  - a. Declare os tipos necessários para a implementação dessa estrutura.
  - b. Escreva um procedimento para cadastrar dois amigos:

**Procedure Relaciona**(int Amigo1, int Amigo2, TipoOrkut\* pOrkut)

Considere que o Amigo1 já está cadastrado no Orkut, enquanto o Amigo2 pode ou não já estar cadastrado.