

Listas ligadas

1. Defina uma estrutura que permita representar uma lista ligada linear simples de números reais.

Escreva subprogramas que desempenhem cada uma das seguintes tarefas. Organize o *main* de forma que seja possível testar cada uma das alíneas.

Sugestão: Comece por organizar o *main* para facilitar os testes dos vários subprogramas.

- a. Criar e devolver uma lista vazia;
- b. Indicar se uma lista é vazia;
- c. Mostrar uma lista;
- d. Mostrar o primeiro elemento de uma lista;
- e. Mostrar o último elemento de uma lista;
- f. Mostrar a cauda (todos os elementos exceto o primeiro) de uma lista;
- g. Determinar o comprimento (número de elementos) de uma lista;
- h. Acrescentar um elemento x no início de uma lista;
- i. Acrescentar um elemento x no fim de uma lista;
- j. Acrescentar um elemento x a seguir a um elemento y da lista;
- k. Acrescentar um elemento x antes de um elemento y da lista;
- l. Construir uma lista a partir de valores inseridos pelo utilizador;
- m. Remover o primeiro elemento de uma lista;
- n. Remover o último elemento de uma lista;
- o. Remover o n -ésimo elemento de uma lista;
- p. Remover a primeira ocorrência do elemento x de uma lista;
- q. Remover a última ocorrência do elemento x de uma lista;
- r. Remover todas as ocorrências do elemento x de uma lista;
- s. Remover as repetições dos elementos de uma lista;
- t. Esvaziar uma lista;
- u. Procurar um valor x numa lista devolvendo um ponteiro para este elemento;
- v. Juntar duas listas numa só;
- w. Copiar uma lista para outra;
- x. Trocar dois elementos de uma lista;
- y. Ordenar uma lista por ordem crescente dos seus valores;
- z. Ordenar uma lista por ordem decrescente dos seus valores, usando um algoritmo diferente do anterior.

2. Considere a estrutura de dados (abstrata) *Pilha* (*stack*, LastInFirstOut).

Elabore subprogramas que executem as operações usadas habitualmente com esta estrutura:

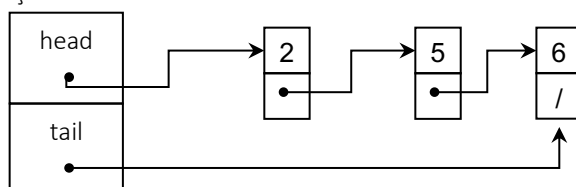
- Criar uma pilha vazia;
- Indicar se uma pilha está vazia;
- Empilhar um elemento na pilha (*push*);
- Retirar um elemento da pilha (*pop*);
- Mostrar o elemento que está no topo da pilha (*top*).

3. Considere a estrutura de dados (abstrata) *Fila* (*queue*, FirstInFirstOut).

Elabore subprogramas que executem as operações usadas habitualmente com esta estrutura:

- Criar uma fila vazia;
- Indicar se uma fila está vazia;
- Inserir um elemento no fim da fila (*juntar*);
- Remover um elemento do início da fila (*remover*);
- Mostrar o elemento que está no início da fila (*frente*).

4. Considere a seguinte representação de uma lista:



- Crie uma nova estrutura lista ajustada a esta nova forma;
- Altere as alíneas a), b), i) e n) do exercício anterior de modo que possam ser usadas nesta nova representação.

5. Refaça o exercício 3. da Ficha de Trabalho n.º 3 usando uma lista ligada de funcionários em vez de uma tabela.

- Defina uma estrutura de dados que, além dos dados de um funcionário, tenha um ponteiro.
- Elabore subprogramas para
 - introduzir os dados de um funcionário na lista;
 - apresentar todos os dados dos funcionários na lista;
 - apresentar todos os dados dos funcionários com salário superior a 500€;
 - procurar e devolver uma estrutura funcionário, usando o seu nome (caso o funcionário não exista, deverá ser devolvido NULL);
 - atualizar os dados de um funcionário (usando o seu número);
 - ordenar a lista por ordem alfabética dos nomes dos funcionários.