

Estruturas de Dados

Universidade da Beira Interior
Departamento de Informática
2010/11

Considere a seguinte estrutura de dados:

```
typedef struct NODO{  
    int chave; //Identificador do nó;  
    ... //campos associados aos dados da lista  
  
    struct NODO *nseg;  
}Nodo;
```

- 1) Implemente uma função para devolver o total de elementos de uma lista.

int totalElementos(Nodo *L);

- 2) Implemente uma função para devolver o total de elementos diferentes de uma lista.

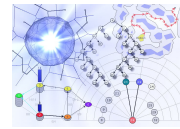
int totalElementosDiferentes(Nodo *L);

a) Sem ordenação

b) Ordenada

- 3) Codifique uma função que permita inserir um novo elemento na primeira posição da lista.

Nodo* insereElementoInicio(Nodo *L, Nodo *nv);



- 4) Codifique uma função que permita inserir um novo elemento na última posição da lista.

Nodo* insereElementoFim(Nodo *L, Nodo *nv);

- 5) Codifique uma função que permita inserir numa posição definida pelo utilizador um novo elemento na lista.

Nodo* insereElemento (Nodo *L, Nodo *nv, int pos);

- 6) Codifique uma função que permita inserir ordenadamente um novo elemento na lista.

Nodo* insereElementoOrdem (Nodo *L, Nodo *nv);

- 7) Verifique se L está ordenada ou não

int listaOrdenada(Nodo *L);

- 8) Faça uma cópia da lista L em uma outra lista L2;

Nodo* copiaLista(Nodo *L);

- 9) Inverta L colocando o resultado em L2;

Nodo* inverteLista (Nodo *L);

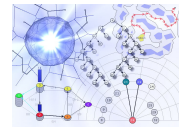
- 10) Inverta L colocando o resultado na própria L;

Nodo* inverteLista (Nodo *L);

- 11) Intercale L1 com a lista L2, gerando a lista L3. Considere que

Nodo* intercalaLista (Nodo *L1, Nodo *L2);

- a) L1, L2 estão ordenadas, bem como a lista resultante L3 deverá estar.



b) L1, L2 não têm ordenação específica, assim como a lista resultante L3.

12) Elimine de L todas as ocorrências de um elemento dado, estando a lista ordenada.

Nodo* intercalaLista (Nodo *L1, int chave);

13) Assumindo que existe um atributo em "Nodo" que guarda um inteiro positivo (int valor), crie uma função que imprima o elemento que ocorre mais vezes na lista.

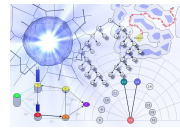
void imprimeFrequente (Nodo *L);

14) Suponha que uma lista simplesmente ligada é utilizada para guardar informação sobre os trabalhos efectuados pelos alunos uma determinada disciplina. Imagine que o número de trabalhos a realizar por cada aluno é variável. A estrutura de dados sugerida é a seguinte:

```
typedef struct TRABALHO{  
    int numero;      //número do aluno;  
  
    int codTrabalho; //identificador do trabalho;  
  
    float nota;      //guarda nota do trabalho [0,20]  
  
    struct TRABALHO *nseg;  
}Trabalho;
```

Implemente funções em linguagem C para cada uma das seguintes operações:

- a) Contar o total de trabalhos feitos por determinado aluno;
- b) Calcular a classificação média de um aluno;
- c) Imprimir o código do trabalho que mais alunos fizeram;



- d) Mostrar uma pauta com as classificações dos alunos a determinado trabalho
 - i) Sem ordenação específica;
 - ii) Mostrando as classificações dos alunos por ordem decrescente;
- e) Contar o número de alunos com classificação média superior a um determinado valor;
- f) Contar o número de alunos que nunca alcançaram uma nota positiva;