

## Exercícios

```
typedef struct node{
    int value;
    struct node*next;
}*link;
```

```
typedef struct node *link;
typedef char* Item;
struct node {
    Item item;
    link next;
};
```

```
a) Item newItem (char* s);
```

b) Crie uma função responsável por criar um novo nodo de uma lista ligada, contendo o Item i e devolvendo o endereço do tipo link.

```
link newNode (Item i);
```

```
c) int itemCompare (Item i1, Item i2);
```

A função `itemCompare` devolve 0 se os itens são iguais. Devolve um valor positivo se  $i1 > i2$  ou um valor negativo caso contrário.

```
d) void showItem (Item i);
```

A função `showItem` mostra o item `i` no standard output.

```
e) void deleteItem (Item i);
```

Liberta a memória associada ao item.

2.3. - Implemente a função void show(link head) que mostra no standard output o conteúdo da lista cujo primeiro elemento é apontado por head. Use as funções do exercício 1.

2.4 - Implemente a função `int search(link head, Item i)` que devolve 1 se o item `i` estiver na lista ou 0 caso contrário.

2.5 - Implemente a função `link remove(link head, Item i)` que remove da lista a primeira ocorrência do Item `i`.

2.6 - Faça um programa que insira numa lista todos os argumentos fornecidos na linha de comandos, e que podem ser obtidos através da tabela argv. De seguida, mostre o conteúdo da lista.

2.7 - Defina que o tipo Item é um inteiro. Modifique as funções do exercício 1 de acordo com a nova definição e resolva os exercícios 2 a 5.

**Nota: Se tiver dificuldade na resolução destes exercícios, tome os exemplos disponibilizados nos slides das aulas teóricas (T17 e T19) como ponto de partida.**