

Lista de Exercícios 12

Para os exercícios abaixo, assuma as seguintes definições:

```
typedef struct integer{  
    int item;  
    Integer *next;  
} Integer;
```

```
typedef Integer * P_i;
```

1. Suponha que *lista* é o (endereço do) primeiro nó de uma lista encadeada que guarda valores inteiros. Escreva uma função que encontre o nó que tem o maior item e transfere esse nó para o fim da lista.
2. Escreva uma função busca-e-remove para listas encadeadas.
3. Escreva uma função que recebe o endereço do primeiro nó de uma lista encadeada, faz uma cópia da lista, e devolve o endereço do primeiro nó da cópia.
4. Escreva uma função que recebe o endereço (do primeiro nó) de uma lista encadeada e retira da lista todos os nós cujo item vale 0. Faça duas versões: na primeira, o primeiro nó da lista é um nó-cabeça (o valor de item é irrelevante); na segunda, a lista não tem nó-cabeça (sua função deve devolver algo nesse caso?).
5. Escreva uma função que retorne o número de nós que possuem item ímpar numa lista encadeada.
6. Escreva uma função que verifique se uma lista encadeada está ordenada.
7. Escreva um procedimento que inverta a ordem dos elementos na lista encadeada, modificando apenas os ponteiros.
8. O que mudaria nas implementações das questões anteriores caso a lista encadeada possuísse uma cabeça (cujo conteúdo de item fosse descartável)?

Para os próximos exercícios, assuma as seguintes definições:

```
typedef struct integer{  
    int item;  
    Integer *next, *prev;  
} Integer;
```

```
typedef Integer * P_i;
```

9. Escreva um procedimento que concatena duas listas duplamente encadeadas.
10. Escreva uma função que remove de uma lista duplamente encadeada o nó apontado por *p*.

11. Escreva uma função que compara os conteúdos de duas listas duplamente encadeadas e retorna uma outra lista apenas com os elementos que estão presentes em uma das listas, mas não na outra. Assuma que não acontecem repetições de elementos internamente a alguma das listas.