

MAC 323 - Estruturas de Dados**Primeiro semestre de 2007****Segunda Lista de Exercícios – Entrega: 17 de abril de 2007**

Para os problemas 1, 2, 3 e 4 nessa lista, estaremos considerando a seguinte estrutura como um nó:

```
struct no{
    struct no* prox;
    int valor;
};
```

1. Crie uma função iterativa que receba um ponteiro para o início de uma lista, inverta-a, e devolva o ponteiro para a nova cabeça. Faça também uma versão recursiva. Atenção, você não pode modificar o valor dos nós, apenas os ponteiros.
2. Crie uma função que receba um ponteiro para o início de uma lista e ordene-a por ordem crescente do conteúdo dos nós. Atenção, você não pode modificar o valor dos nós, apenas os ponteiros.
3. Crie uma função que receba dois ponteiros para listas distintas e ordenadas, e devolva um ponteiro para uma terceira lista gerada pela intersecção dos elementos das duas listas iniciais. As listas originais devem ser destruídas. Você não pode alocar novos nós para fazer a tarefa.
4. Crie uma função que receba dois ponteiros para listas distintas e ordenadas, e devolva um ponteiro para uma terceira lista gerada pela união dos elementos das duas listas iniciais. As listas originais devem ser destruídas. Você não pode alocar novos nós para fazer a tarefa.
5. Faça uma função que receba ponteiros para duas listas que representam polinômios e devolva o produto entre eles. Considere que os nós da lista estão ordenados em ordem crescente do expoente de x . Você não pode alocar mais nós do que o número de nós que precisará para representar o polinômio resultante.
6. Faça uma função que receba ponteiros para duas representações de matrizes esparsas por listas encadeadas e devolva o produto delas. Considere que a multiplicação é sempre possível (isto é as matrizes são de tamanho $A \times B$ e $B \times C$).

Algumas notas finais:

- Para todos os problemas você deve alocar os nós através da função 'AlocaNo' e liberá-los com a 'LiberaNo'. Considere que você sempre consegue alocar os nós.
- As listas a serem utilizadas serão sempre simples, sem cabeça. Considere que nunca serão passadas listas vazias como parâmetros.
- Em momento algum será necessário fazer 'malloc's, ou tratamento de erro de entrada.
- Se precisar usar alguma outra estrutura de dados como pilha, fila ou fila dupla, não precisa implementá-la.
- Coloque na resposta somente a função pedida, e as auxiliares que ela possa precisar. Não é necessário o código de todo o programa.
- Faça comentários em seu código para explicar o que não for óbvio. Exemplo: você não precisa explicar quando está alocando um novo nó. Mas é bom que explique quando está procurando por algo na lista.
- Para todos os exercícios você terá que fazer, pelo menos, um pseudo-código. Somente explicar o que seu programa fará não serve.
- Entregue sua lista preferencialmente em papel, em vez de usar a submissão online. Como sabem o Instituto está com alguns problemas na parte de impressão e, às vezes, é difícil conseguir imprimir as listas para corrigi-las.