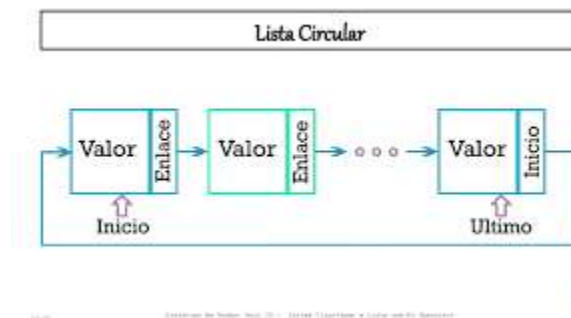


# Estruturas de Dados I

Prof. Roney Pignaton da Silva

## Exercícios sobre Listas, Pilhas e Filas

1. Considere uma lista simplesmente encadeada circular sem cabeça, conforme mostrado abaixo.



Esta lista mantém um ponteiro para o primeiro e último elemento da lista e não possui cabeça.

Implemente um programa que contém as seguintes funções:

- a) Inserção no início da lista.
  - b) Remoção (free) do primeiro elemento da lista.
  - c) Inserção na última posição da lista.
  - d) Remoção (free) do último elemento da lista.
  - e) Remoção (free) de uma chave informada.
  - f) Impressão da lista.
2. Considere o seguinte método para codificar mensagens de texto:
    - Primeira etapa- todas sequências não-vogais, incluindo o espaço em branco e os caracteres de pontuação são invertidas;
    - Segunda etapa- a mensagem inteira resultante é invertida dando origem à mensagem codificada.

Exemplo: Dada a mensagem:

ESTRUTURAS DE DADOS E MUITO LEGAL.

Após a primeira etapa, teremos:

ERTSUTURAD SED ADO SEM UITOL EGA.L

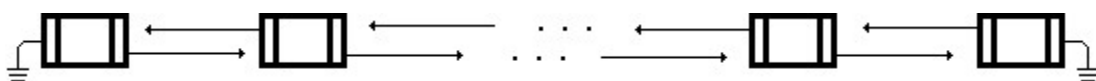
E depois da segunda etapa teremos:

L.AGE LOTIU MES ODA DES DARUTUSTRE

Modele o TAD List mostrado abaixo e implemente uma função para decodificar as mensagens assim codificadas. Use uma lista duplamente ligada para representar a mensagem onde cada elemento contém um caractere no campo de informação.

**Assinatura da função:** List\* decodifica(List \*lst)

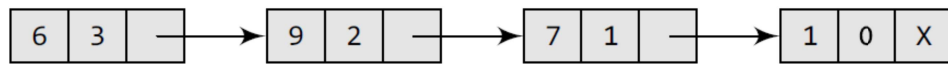
**Modelo da lista:**



Sugestão: Implemente uma sub-função “List \*inverte(List \*p,List \*q)” que inverte uma lista entre os limites apontados por p e q.

3. Considere que uma lista simplesmente ligada é usada para representar um polinômio, conforme o exemplo abaixo:

$$6x^3 + 9x^2 + 7x + 1.$$



Como se pode ver, cada termo individual do polinômio consiste de duas partes: 1) coeficiente e 2) a potência do termo.

Nesse exemplo, 6, 9, 7, and 1 são os coeficientes dos termos de potência 3, 2, 1, e 0, respectivamente.

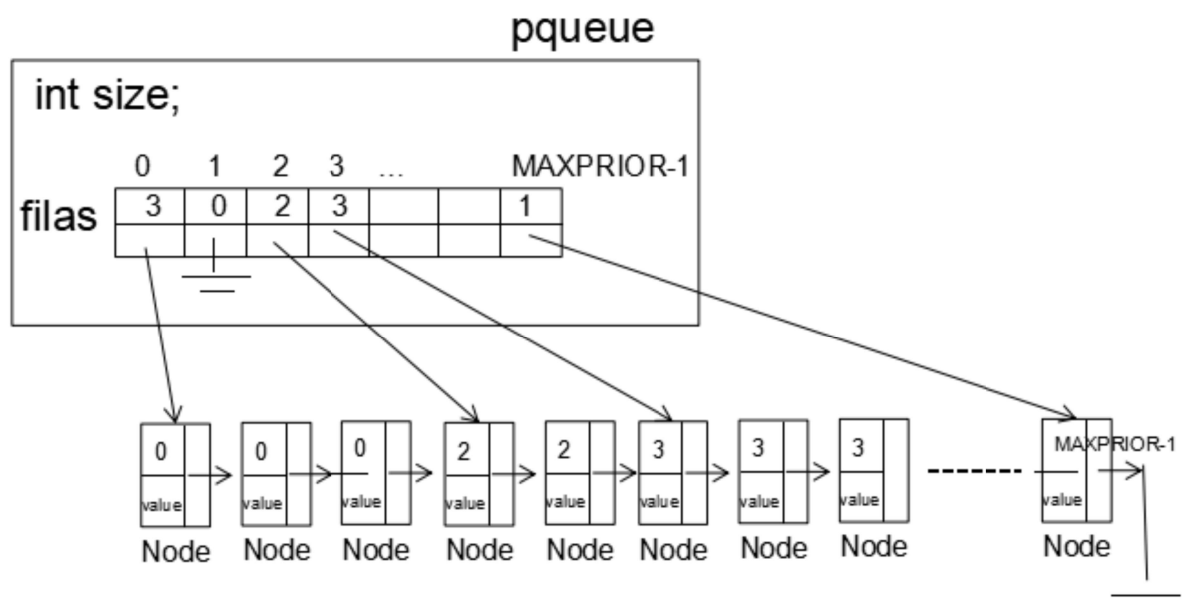
Considerando o exposto e que a estrutura de dados usada para representar o polinômio é uma lista sem cabeça cuja estrutura é mostrada abaixo:

```
struct node
{
    int num;
    int coeff;
    struct node *next;
};
```

implemente as funções que realizam as seguintes operação sobre polinômios:

- 1) struct node \*add\_poly(struct node \*start1, struct node \*start2, struct node \*start3): função que soma dois polinômios (start1 e start2) retornando o polinômio resultado (start3)
- 2) struct node \*sub\_poly(struct node \*start1, struct node \*start2, struct node \*start3): função que subtrai dois polinômios (start1 e start2) retornando o polinômio resultado (start3)
- 1) struct node \*mult\_poly(struct node \*start1, struct node \*start2, struct node \*start3): função que multiplica dois polinômios (start1 e start2) retornando o polinômio resultado (start3)
- 2) struct node \*div\_poly(struct node \*start1, struct node \*start2, struct node \*start3): função que divide dois polinômios (start1 e start2) retornando o polinômio resultado (start3)

4. Considere uma implementação de uma fila de prioridades conforme a estrutura mostrada abaixo:



Considerando que esta fila de prioridades mantém um apontador no campo filas para cada sublista representando os elementos pertencentes às diferentes prioridades, implemente:

1) a estrutura de dados pqueue;

2) as seguintes funções

a) void insertPQueue(Pqueue \*pq, int prior, int inf): função que insere um elemento de informação (inf) na fila (porque), em sua sublista definida pela prioridade (prior)

b) removePQueue(Pqueue \*pq, int prior, int quant): função que retira uma quantidade (quant) de elementos de informação (inf) da fila (porque), removendo tais elementos da sublista definida pela prioridade (prior)