

**Observação**

Para os exercícios 27 a 32 implemente o tipo abstrato de dados lista com as seguintes operações:

- Criar lista vazia;
- Inserir no fim; (add)
- Inserir no início; (insert)
- Inserir na i-ésima posição (posição e item são passados por parâmetro);
- Excluir na i-ésima posição;
- Excluir um item pela chave; (remove)
- Localizar/Pesquisar um elemento (search)

**EXERCÍCIO 27**

Implemente uma lista estática (array).

**EXERCÍCIO 28**

Implemente uma lista simplesmente encadeada estática (array).

**EXERCÍCIO 29**

Implemente uma lista simplesmente encadeada com tail. Além das operações solicitadas, implementar:

- método para fazer uma cópia da lista;

-----  
OBS: Implemente os exercícios 30, 31 e 32 usando programação genérica.  
-----

**EXERCÍCIO 30**

Implemente uma lista simplesmente encadeada com sentinela. Além das operações solicitadas, implementar:

- método para concatenar (Merge)
- método para ordenar os itens da lista em ordem ascendente.

**EXERCÍCIO 31**

Implemente uma lista duplamente encadeada com sentinelas. Além das operações solicitadas, implementar:

- método para concatenar (Merge)
- método para dividir uma lista em várias
- método para ordenar os itens da lista em ordem ascendente.

**EXERCÍCIO 32**

Implemente uma lista circular duplamente encadeada

**EXERCÍCIO 33**

---

Considerando listas simplesmente encadeadas (exercício 29), implemente:

- um método ao receber uma lista por parâmetro, retorne uma lista que é a união sem repetição das duas outras listas.
- um método ao receber uma lista por parâmetro, retorne uma lista que é a interseção das duas outras listas.

### EXERCÍCIO 34

Desejamos manipular polinômios do tipo  $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ . Tais polinômios podem ser representados por listas duplamente encadeadas onde cada elemento da lista possui dois campos: um para o coeficiente que é um número real, um para o expoente que é um número inteiro. Implemente uma classe que represente um polinômio e que tenha métodos para somar dois polinômios e derivar um polinômio.

### EXERCÍCIO 35

Construa um procedimento para inverter a ordem de uma lista simplesmente encadeada utilizando apenas duas variáveis auxiliares (não é permitido o uso de listas auxiliares). É possível construir o procedimento utilizando apenas uma variável auxiliar? Se sim, como?

### EXERCÍCIO 36

Mostre como implementar uma pilha utilizando-se de duas filas.

### EXERCÍCIO 37

Faça um programa que cadastre em uma pilha vários números. A entrada deles será finalizada com a digitação de um número menor que 0. Posteriormente o programa deve gerar duas filas, a primeira com os números pares e a segunda com os números ímpares. A saída do programa deve apresentar a pilha digitada e as filas geradas. Caso alguma das filas seja vazia, deve-se mostrar a mensagem.

### EXERCÍCIO 38

DEQUE é uma fila de duas pontas, cuja estrutura de dados consiste de uma lista na qual as seguintes operações são permitidas:

- Push(A) Insere o elemento A no início da deque.
- Pop() Remove o elemento que está no início da deque.
- Inject(A) Insere o elemento A no final da deque.
- Eject() Remove o elemento que está no final da deque.

A fim de implementar as operações acima em tempo constante, qual a melhor opção: lista sequencial, lista simplesmente encadeada ou lista duplamente encadeada? A partir de sua decisão, crie uma classe que represente a estrutura de dados e as operações acima.

### EXERCÍCIO 39

Duas pilhas podem ser implementadas em um único array A da seguinte forma: A primeira pilha cresce a partir do início do array para direita e a segunda cresce a partir do final do array para a esquerda, ou seja, as pilhas crescem uma em direção a outra. Escreva os seguintes procedimentos:

- Push(A,i), onde A é o novo elemento a ser inserido na pilha i = 1,2.
- Pop(i) que retorna o elemento que está no topo da pilha i = 1,2.

### EXERCÍCIO 40

Utilizando a implementação de listas através de arranjos (Lista sequencial), implemente um procedimento para inserir um item em uma determinada posição da lista.

---

**EXERCÍCIO 41**

Implemente um procedimento RemovePar que, dada uma lista autorreferenciadas remove todos elementos cuja chave é par.

**EXERCÍCIO 42**

Implemente um procedimento FuraFila que, dada uma fila implementada com apontadores insere um elemento na primeira posição da fila.

**EXERCÍCIO 43**

Utilizando as operações de manipulação de pilhas vistas em sala, uma pilha auxiliar e uma variável do tipo TipoItem, escreva um procedimento que remove um item com chave c de uma posição qualquer de uma pilha. Note que você não tem acesso à estrutura interna da pilha (topo, item, etc), apenas às operações de manipulação.

**EXERCÍCIO 44**

Agora escreva uma função para trocar os elementos m e n de uma lista duplamente encadeada (m e n podem ser chaves ou mesmo ponteiros para os elementos).

BOM ESTUDO!