

**ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI**  
**ENGENHARIA DA COMPUTAÇÃO - ECOMP**  
**ESTRUTURAS DE DADOS - 2019.2**

**Lista de exercícios 02 - Listas encadeadas**  
**23 de agosto de 2019**

**(ESCOLHA 05 QUESTÕES PARA RESPONDER)**

**1)** Crie uma lista que atua sobre um TAD. Para essa questão, considere que queremos modelar uma lista encadeada de clientes de uma dada empresa. Para isso, modele o dado Empresa (nomeEmpresa, listaClientes) e Cliente (nome ,cpf). Construa os seguintes métodos:

- a) Adicionar um cliente a lista (um único cliente por cpf);
- b) Imprimir todos os dados de todos os clientes;
- c) Verificar se um cliente está contido na lista;
- d) Exibir o número de clientes cadastros.

**2)** Dada uma lista estática, crie os seguintes métodos:

- a) Add, remove, size e print. O add deve aumentar o array caso necessário.
- b) Remover todas as instâncias de um elemento na lista.
- c) Recebendo dois elementos, trocá-los de lugar na lista. caso haja repetição, trocar apenas o primeiro.
- d) Concatenar uma lista recebida à lista da classe.

**3)** Implemente uma lista encadeada circular que modela um TAD. Para essa questão, desenvolva os seguintes métodos:

- a) Recebe duas listas e verifica se a sequência de dados da segunda lista está contida na primeira. Caso esteja, remova a sequência de valores da primeira lista.

**Exemplos:**

lista1: 2, 5, 7, 10, 1  
lista2: 10, 2, 5

Saída: false

lista1: 9, 8, 5, 10, 12, 13  
lista2: 10, 12, 13

Saída: true  
Nova lista1: 9, 8, 5

lista1: a, b, c, d, e  
lista2: d, e, a

Saída: true  
Nova lista1: b, c

**ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI**  
**ENGENHARIA DA COMPUTAÇÃO - ECOMP**  
**ESTRUTURAS DE DADOS - 2019.2**

b) Dada duas listas L1 e L2, adicionar intercaladamente seus elementos em uma outra lista L3;

4) Utilizando uma lista simplesmente ligada que armazena Strings em seus nós, efetue a inversão da lista conforme os exemplos abaixo. **Faça a inversão através dos campos de ligação, e NÃO dos campos de informação.**

Exemplos:

Lista original	Lista invertida
1→2→2→3	3→2→2→1
m→a→d→a→m→i→m→a→d→a→ m	m→a→d→a→m→i→m→a→d →a→m
d→e→v→e→l→o→p→e→r	r→e→p→o→l→e→v→e→d
Eu→amo→ED	ED→amo→Eu

5) Dada uma lista ligada onde todo nó representa uma lista ligada e contém dois ponteiros dos tipos:

- i) Ponteiro para o próximo nó na lista principal
- ii) Ponteiro para uma lista ligada onde esse dado nó é a cabeça

Implemente 3 métodos:

- a) addListaPrincipal(T e), que deve inserir o valor passado como argumento ordenadamente na lista principal.
- b) addOutraLista(T e, int indexListaPrincipal), que inserirá o valor passado como argumento ordenadamente na lista secundária de index passado como argumento.
- c) planificar(), que planificará a lista em uma lista ligada simples que deve ser ordenada.

Ex.: A seguinte lista:

```

5 -> 10 -> 19 -> 28
|      |      |      |
7      20     22     35
|      |      |      |
8      50     40
|      |      |      |
30     45

```

Ficará da seguinte forma ao se planificar:

5->7->8->10->19->20->22->28->30->35->40->45->50.

**ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI**  
**ENGENHARIA DA COMPUTAÇÃO - ECOMP**  
**ESTRUTURAS DE DADOS - 2019.2**

**obs.:** A lista não aceitará números repetidos na mesma lista. Ou seja, caso a `addListaPrincipal` ache um valor repetido, não deverá duplicá-lo. Da mesma forma, `addOutraLista` não deverá repetir o mesmo número, caso o haja no `index` passado como argumento.

**obs.2:** `planificar()` deverá modificar a própria lista.

**6)** Implemente a TAD lista, dinâmica, duplamente ligada, não ordenada de String e desenvolva os seguintes métodos:

- a) Adicionar um elemento da lista;
- b) Remover um elemento da lista;
- c) Imprimir todos os elementos da lista;
- d) Buscar um dado elemento com base no índice passado como argumento;
- e) Receber duas listas distintas e retornar outra lista com os elementos em comum das listas recebidas.