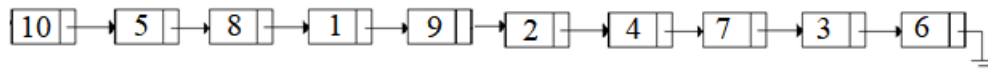
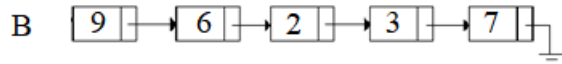
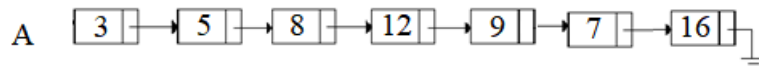


1) Dada a Lista L abaixo, fazer:



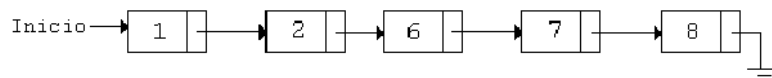
- Ordenar a lista (Usar outros TADs é permitido)
- Implementar, em Java, um projeto que inicie a Lista L como apresentada acima e, faça sua ordenação e exiba ordenada. Permitido usar bibliotecas de outros TADs

2) Dadas as listas A e B, faça:



- a) Criar a sequência de passos para criar a lista I e a lista U onde:
 - i. I é a lista com a intersecção das listas A e B
 - ii. U é a lista com a união de A e B
- b) Fazer, em Java, um novo projeto que inicialize as listas A e B e tenha métodos que gere e imprima as listas I e U.

3) Dada uma lista inicial L, conforme o exemplo abaixo, demonstre, com operações de lista, como ficará L ao final das aplicações dos passos, como determinado abaixo:



Baseado no vetor abaixo:

Para cada posição do vetor:

- Se o valor já existe na lista, inserir no fim
- Senão, se o valor for par, inserir na posição 2
- Senão, se o valor for ímpar, inserir na posição 1

Vetor

3	3	9	6	9	8	9	5	7	10	4	8	10	8
---	---	---	---	---	---	---	---	---	----	---	---	----	---

4) Considerando o projeto que gerou a biblioteca ListInt no exercício 1, fazer uma modificação para criar a biblioteca SetInt. Set é uma coleção de dados semelhante a uma Lista Encadeada Simples, no entanto, os métodos de adição de um novo elemento precisam ter uma verificação extra, que consiste como segue:

Um valor não pode ser inserido na lista se já houver um valor igual adicionado na lista

Criar um novo Projeto Java SetInt semelhante ao projeto de Lista de Inteiros, aplicar a verificação de elementos iguais em todos os métodos de adição e criar a biblioteca com o JAR SetInt

5) Considerando a biblioteca ListaString criada no exercício 1, fazer uma aplicação Java que simule uma playlist de músicas em um aplicativo gratuito. O tal aplicativo Vitufy, por ser gratuito, a partir de uma lista, só permite a execução das músicas na ordem em que forem inseridas, não permitindo avançar nem retornar.

Criar um objeto Musica (String nome; String nomeArtista; int duracao)

Criar uma classe PlayerController no Package controller, que tenha alguns métodos:

- adicionaMusica(Lista lista,String musica):void, que receberá uma String no formato: Nome_da_musica;nome_artista;duracao (obrigatoriamente dessa forma), fazer o split e gravar em um objeto Musica e, se a lista estiver vazia, adicione na primeira posição. A partir do segundo elemento, todas as outras músicas devem entrar no final da lista;
- removeMusica(Lista lista, int posição): void, permitindo que o usuário remova qualquer música da lista. Se a lista estiver vazia, uma exceção deve ser mostrada;
- executaPlaylist(Lista lista): void, que exibirá, do primeiro ao último elemento, no formato [Música: nome da música – Artista: nome artista – Duração: duracao]. Se a lista estiver vazia, uma exceção deverá ser lançada.

Bônus: Gerar um sleep para simular a música tocando. Por não ser uma classe tipo Thread, deve-se usar Thread.sleep(tempo); para conseguir o mesmo efeito.

No Package view, criar uma classe chamada Principal, e, na sua Main, inicialize uma Lista Encadeada e oferecer ao usuário a opção de inserir ou remover músicas na playlist e executar a playlist. No caso da inserção de músicas, deve-se pedir, ao usuário, que digite separadamente cada campo e monte uma String que será mandada ao método adicionaMusica na classe PlayerController. A aplicação deve ficar ativa até que o usuário digite uma opção de sair.