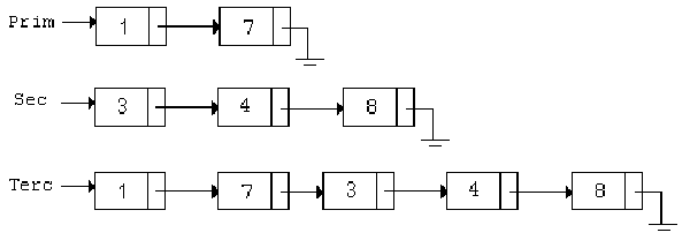
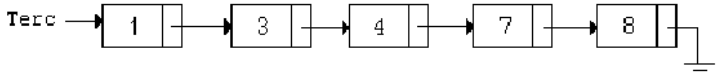
1. Fazer um teste de mesa do algoritmo abaixo, deixando explícito os valores de saída em console e a estrutura da Lista

```
Lista l = new Lista();
int[] vetor = {100, 200, 1, 50, 39, 44, 25, 16, 99, 45, 33, 18, 102, 92};
int tamanhoVetor = vetor.length;
Para (i = 0 ; i < tamanhoVetor ; i++) {
        Se (listaVazia() ==  verdadeiro) {
                l.addFirst(vetor[i] * 8);
        } Senao Se (vetor[i] mod 5 == 0 || l.size() > 10) {
                l.addLast(vetor[i] + 10);
        } Senao Se (l.size() >= 2) {
                l.add(vetor[i] * 3, 1);
        } Senao {
                l.addFirst(vetor[i]);
        }
}
int tamanho = l.size();
Enquanto (! ListaVazia()) {
        Se (tamanho > 10) {
                escreva(l.get(0));
                removeFirst();
        } Senao Se (tamanho > 1) {
                escreva(l.get(tamanho – 1));
                removeLast();
        } Senao {
                escreva(l.get(0));
                removeFirst();
        }
        int tamanho = l.size();
}
```

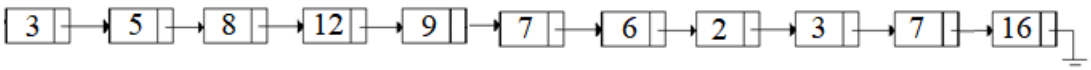
2. Considerando as listas já criadas, apresente a sequência lógica do teste de mesa para:
   a) Dadas as listas Prim e Sec, fazer a sequência de operações que gere a lista Terc



   b) Considerando a lista Terc gerada, fazer a sequência de passos que ordene a lista Terc do menor para o maior



3. Dada a Lista L abaixo, fazer:



   a) Simule, em teste de mesa, a sequência de passos para exibir os 2 maiores valores da lista
   b) Fazer, em Java, um novo projeto que inicialize a lista L, com os dados acime e implemente operações que permitam determinar os 2 maiores valores da lista. O uso de bibliotecas de outros TADs é permitido.

3. Fazer o teste de mesa considerando o vetor

25	42	23	74	80	77	13	41	59	35	68	53	75	84	44	94	39	71	88

A partir de uma Lista L iniciada:

Para cada elemento do vetor:

- Se a lista estiver vazia, inserir, no início, o dobro do valor
- Senão, se a lista tiver menos de 3 elementos, inserir, no fim, a metade do valor (Inteiro)
- Senão, se a lista tiver mais de 10 elementos, inserir, na posição 5, o triplo do valor
- Senão, inserir, na posição 1, o valor

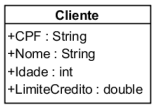
Ao terminar de percorrer o vetor, fazer:

Enquanto a lista tiver elementos:

- Se a lista tiver mais de 10 elementos, remover a posição 3 e exibir
- Senão, se a lista tiver mais de 5 elementos, remover do fim e exibir
- Senão, se a lista tiver mais de 3 elementos, remover a posição 1 e exibir
- Senão, remover do início e exibir

5. A Empresa XYZ quer fazer um ajuste no seu cadastro de clientes trocando o que está em um único arquivo para diversos arquivos, a fim de facilitar a busca quando o Cliente fizer uma compra:

1) Criar um objeto model com os atributos:



2) Criar, manualmente, um arquivo texto chamado cadastro.csv na pasta C:\TEMP (Caso a pasta não exista, cria-la manualmente, também)
3) Preencher os arquivos com os dados oferecidos no final do enunciado
4) Criar uma classe controller, chamada ModificacaoCadastroController:
   - Um método, private novoArquivo(Lista l, String nomeArquivo) : void, que deverá:
     a. Inicializar um new File com o caminho “C:\\TEMP” e o nome do Arquivo passado como parâmetro
     b. Inicializar um contador
     c. Inicializar um StringBuffer
     d. Percorrer a lista de Objetos
     e. Para cada elemento da lista, criar uma String no formato csv, igual ao arquivo cadastro.csv (Não esquecendo a quebra de linha ao final da String)
     f. Gravar o buffer no novo arquivo
   - Um método, public novoCadastro(int idadeMin, int idadeMax, Double limiteCredito) : void, que deverá:
     a. Inicializar uma lista de clientes
     b. Percorrer o arquivo cadastro.csv
     c. Para cada linha do arquivo, fazer uma operação de split e armazenar em um novo objeto cliente, os valores
     d. Verificar se a idade está entre os valores de idade passados como parâmetro(min e max) e o limite de crédito seja maior que o limite de crédito passados como parâmetro. Em caso positivo, adicionar à lista;
     e. Chamar o método de escrita de novo arquivo, passando a lista como parâmetro e o nome do novo arquivo como parâmetros. O nome do novo arquivo deve ser: “Idade “+idade+” limite”+limiteCredito;
5) Criar uma classe view (Principal.java) que, na main:
   a. Chame o método novoCadastro da classe ModificacaoCadastroController 3 vezes:
      i. Para idade entre 41 e 60 e limite de Crédito acima de 8000.00
      ii. Para idade entre 31 e 40 e limite de Crédito acima de 5000.00
      iii. Para idade entre 21 e 30 e limite de Crédito acima de 3000. 00

Obs.: Linhas que estejam no arquivo cadastro.csv e não forem contemplados nas condições serão desconsideradas pela empresa

Cadastro.csv (CPF, Nome, Idade, LimiteCredito)

```
54707521304;Cliente A;45;11108,00
19003628372;Cliente B;41;9756,00
54446710134;Cliente C;33;7217,00
93126907239;Cliente D;40;14353,00
34935227995;Cliente E;30;7342,00
26679245568;Cliente F;53;9491,00
47817135372;Cliente G;31;5244,00
88160205180;Cliente H;31;12817,00
13667799861;Cliente I;28;12686,00
27977151616;Cliente J;21;11092,00
56634688310;Cliente K;18;7697,00
77407950202;Cliente L;45;8877,00
76113821167;Cliente M;26;3865,00
60505933383;Cliente N;18;3978,00
38134578281;Cliente O;26;9574,00
39557597722;Cliente P;27;6476,00
43535332258;Cliente Q;34;3776,00
13897841035;Cliente R;32;7964,00
22264929711;Cliente S;34;4204,00
24246599452;Cliente T;31;4869,00
94461659861;Cliente U;22;12770,00
95029452114;Cliente V;49;11877,00
61587656636;Cliente W;43;4802,00
33660496913;Cliente X;46;8208,00
73014492090;Cliente Y;19;6177,00
46043531381;Cliente Z;20;10641,00
```