Instruções para a entrega: fazer os exercícios a seguir e mostrar para o professor no computador na aula do dia 22/mar. A entrega pode ser em dupla, alunos ausentes não terão a nota considerada.

Considere as classes a seguir nos exercícios. A classe Lista implementa uma lista encadeada **circular** e todos os exercícios deverão utilizar lista circular.

```
public class No {
  int conteudo;
  No proximo;
}
```

```
public class Lista {
  No inicio;

  void add(int nro) {
    No no = new No();
    no.conteudo = nro;
    if( inicio == null ){
      no.proximo = no;
      inicio = no;
    }
    else{
      No ultimo = inicio;
      while( ultimo.proximo != inicio ){
        ultimo = ultimo.proximo;
      }
      no.proximo = ultimo.proximo;
      ultimo.proximo = no;
    }

  }

  void print() {
    if( inicio == null ){
      System.out.println("\nLista vazia");
    }
    else{
      System.out.println();
      No ultimo = inicio;
      do{
        System.out.print(ultimo.conteudo+" ");
        ultimo = ultimo.proximo;
      }while( ultimo != inicio );
    }
  }
}
```

Exercício 1 – Alterar o método ***add(nro:int)*** para manter os elementos em ordem crescente de valor.

Exercício 2 – Programar na classe ***Lista*** um método de nome ***sum(lista:Lista)*** que recebe outra lista e adiciona os elementos dela na lista atual mantendo a regra do Exercício 1, isto é, a lista final precisar estar ordenada.

Observação: utilize o método add(nro:int) programado no Exercício 1 para adicionar os novos elementos mantendo a ordem.

Exemplo de teste do Exercício 2.

```
public class Principal {
  public static void main(String[] args) {
      Lista a = new Lista();
      a.add(5);
      a.add(2);
      a.add(8);
      a.add(6);
      a.add(7);
      a.print();
      Lista b = new Lista();
      b.add(4);
      b.add(3);
      b.add(11);
      b.add(9);
      b.print();
      a.sum(b);
      a.print();
      Lista c = new Lista();
      a.sum(c);
      a.print();
      c.add(10);
      a.sum(c);
      a.print();
  }
}
```

```
Console
2 5 6 7 8
3 4 9 11
2 3 4 5 6 7 8 9 11
2 3 4 5 6 7 8 9 11
2 3 4 5 6 7 8 9 10 11
```

Exercício 3 – Programar na classe ***Lista*** um método de nome ***remove():No*** que retira e retorna o nó que está no início da lista.

Exemplo de teste do Exercício 3.

```
public class Principal {
  public static void main(String[] args) {
    No no;
    Lista a = new Lista();
    no = a.remove();
    System.out.println("Remover: " + no);
    a.add(5);
    no = a.remove();
    System.out.print(
                  "Remover: " + no.conteudo);
    a.add(2);
    a.add(8);
    a.add(6);
```

```
    a.add(7);
    a.print();
    no = a.remove();
    System.out.print(
                  "Remover: " + no.conteudo);
    a.print();
  }
}
```

Console

```
Remover: null
Remover: 5
2 6 7 8
Remover: 2
6 7 8
```

Exercício 4 – Programar na classe ***Lista*** o método ***interrupt(nro:int)***. Ele interrompe a lista no elemento que tiver o conteúdo passado como parâmetro, mas lembre-se que a lista ainda deverá ser circular.

Exemplo de teste do Exercício 4.

```
public class Principal {
  public static void main(String[] args) {
    Lista a = new Lista();
    a.interrupt(5);
    a.add(5);
    a.interrupt(5);
    a.add(9);
    a.add(11);
    a.add(4);
    a.add(2);
    a.add(3);
    a.add(8);
    a.add(1);
    a.print();
    a.interrupt(4);
    a.print();
  }
}
```

Console

```
1 2 3 4 5 8 9 11
1 2 3 4
```

Exercício 5 – Programar na classe ***Lista*** o método ***split(nro:int):Lista***. Ele quebra uma lista em duas no elemento que tiver o conteúdo passado como parâmetro.

Exemplo de teste do Exercício 5.

```
public class Principal {
  public static void main(String[] args) {
    Lista a = new Lista();
    Lista b = a.split(5);
    b.print();
    a.add(5);
    b = b.split(5);
    b.print();
    a.add(9);
    a.add(11);
    a.add(4);
    a.add(2);
    a.add(3);
    a.add(8);
    a.add(1);
    a.print();
    b = a.split(4);
    a.print();
    b.print();
  }
}
```

Console

```
Lista vazia
Lista vazia
1 2 3 4 5 8 9 11
1 2 3 4
5 8 9 11
```