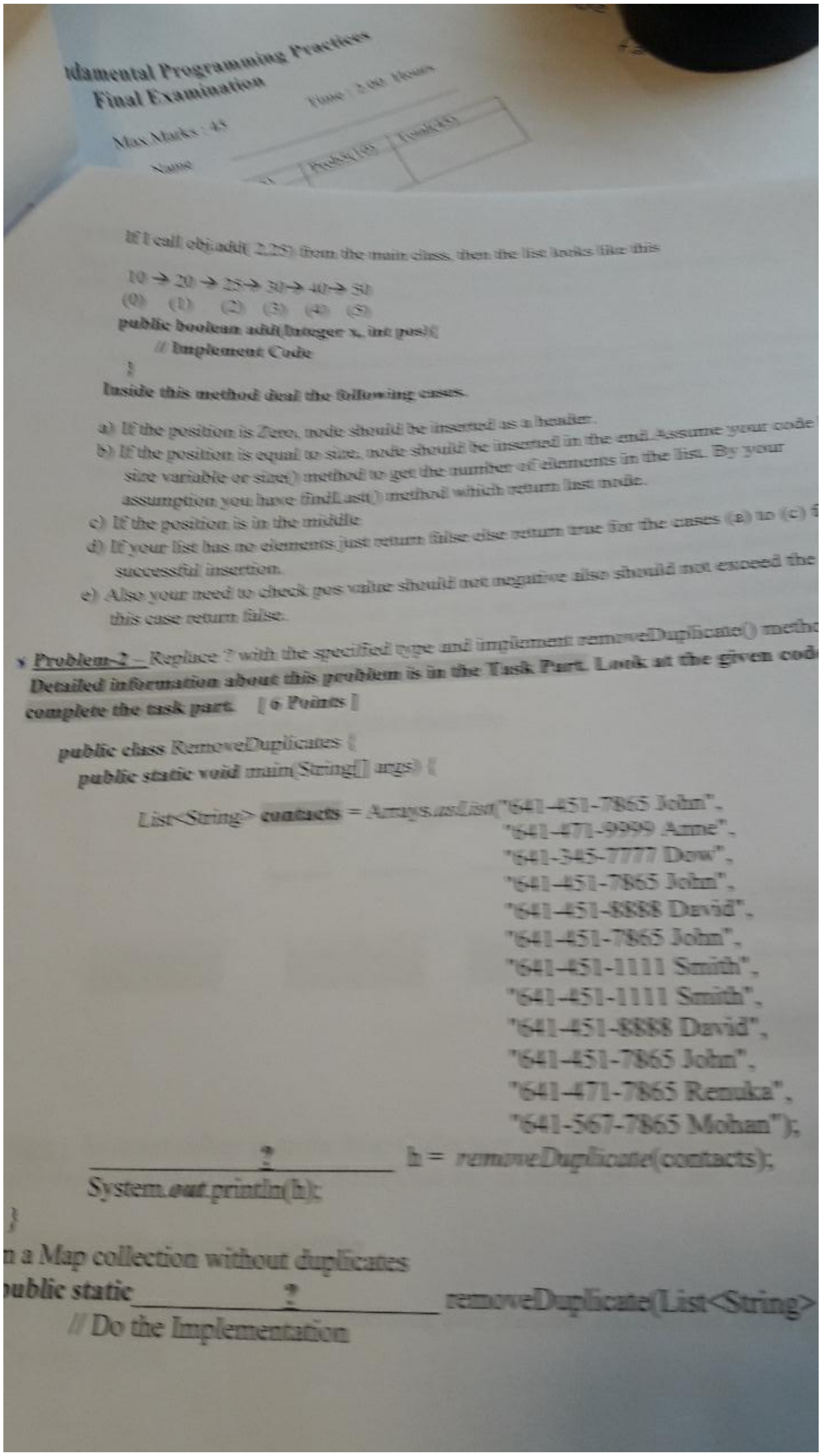
QUESTION NO 1 Implementation of Enqueue using ArrayList .



ndamental Programming Practices
Final Examination

Max Marks : 45 Time : 2.00 Hours

Name

Prob5(10)	Total(45)

If I call obj.add(2,25) from the main class, then the list looks like this

```
10 → 20 → 25→ 30→ 40→ 50
(0)   (1)   (2)  (3)  (4)  (5)
public boolean add(Integer x, int pos){
    // Implement Code
}
```

Inside this method deal the following cases.

a) If the position is Zero, node should be inserted as a header.
b) If the position is equal to size, node should be inserted in the end. Assume your code
size variable or size() method to get the number of elements in the list. By your assumption you have findLast() method which return last node.
c) If the position is in the middle
d) If your list has no elements just return false else return true for the cases (a) to (c) f
successful insertion.
e) Also your need to check pos value should not negative also should not exceed the
this case return false.

* <u>**Problem-2 –**</u> Replace ? with the specified type and implement removeDuplicate() metho
Detailed information about this problem is in the Task Part. Look at the given cod complete the task part. [**6 Points**]

```
public class RemoveDuplicates {
  public static void main(String[] args) {
        List<String> contacts = Arrays.asList("641-451-7865 John",
                                          "641-471-9999 Anne",
                                          "641-345-7777 Dow",
                                          "641-451-7865 John",
                                          "641-451-8888 David",
                                          "641-451-7865 John",
                                          "641-451-1111 Smith",
                                          "641-451-1111 Smith",
                                          "641-451-8888 David",
                                          "641-451-7865 John",
                                          "641-471-7865 Renuka",
                                          "641-567-7865 Mohan");
  _____________?_________ h = removeDuplicate(contacts);
  System.out.println(h);
}
n a Map collection without duplicates
ublic static_____________?_____________ removeDuplicate(List<String>
    // Do the Implementation
```

lamental Programming Practices
Final Examination

Max.Marks : 45 Time : 2.00 Hours

Prob3(10)	Total(45)

// Expected Output

```
{641-567-7865=Mohan, 641-451-1111=Smith, 641-345-7777=Dow, 641-471-9999=Aame,
641-451-8888=David, 641-451-7865=John, 641-471-7865=Remika}
```

Task Part

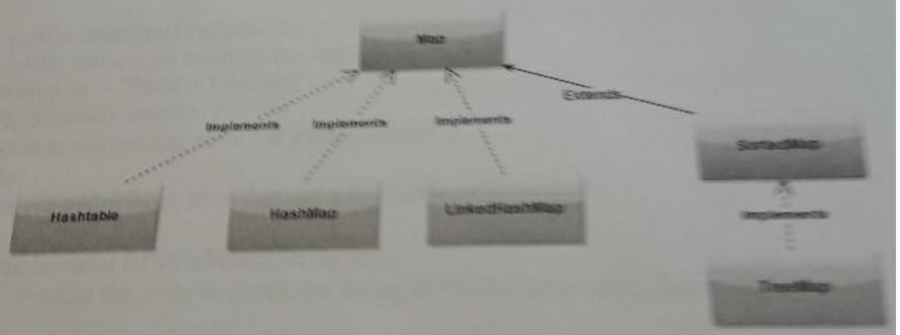
Observations from the above code: There are some duplicate elements in the List contacts, because List interface allows duplicates. From the given contacts list take a sample input "641-451-7865 John", there are two fields in this given input. First input "641-451-7865" is contact number, another input "John" is a name.

Perform the following to remove the duplicates.

a. Replace ? with suitable Map Interface which accepts <Key,Value> pair and does not allo duplicate values also does not allow null key and null values. Then choose the right data type for Key and Value.

b. removeDuplicate() method, which accepts a contacts list as an argument and return Ma collection as a result without duplicates. Write a code inside this method which inserts k and value to your Map collection. Here from the input take first word as a Key and nex word as a Value. Example "641-451-7865 John", 64-451-7865 is a Key and John is va Then store this Key and Value in to the Map collection.

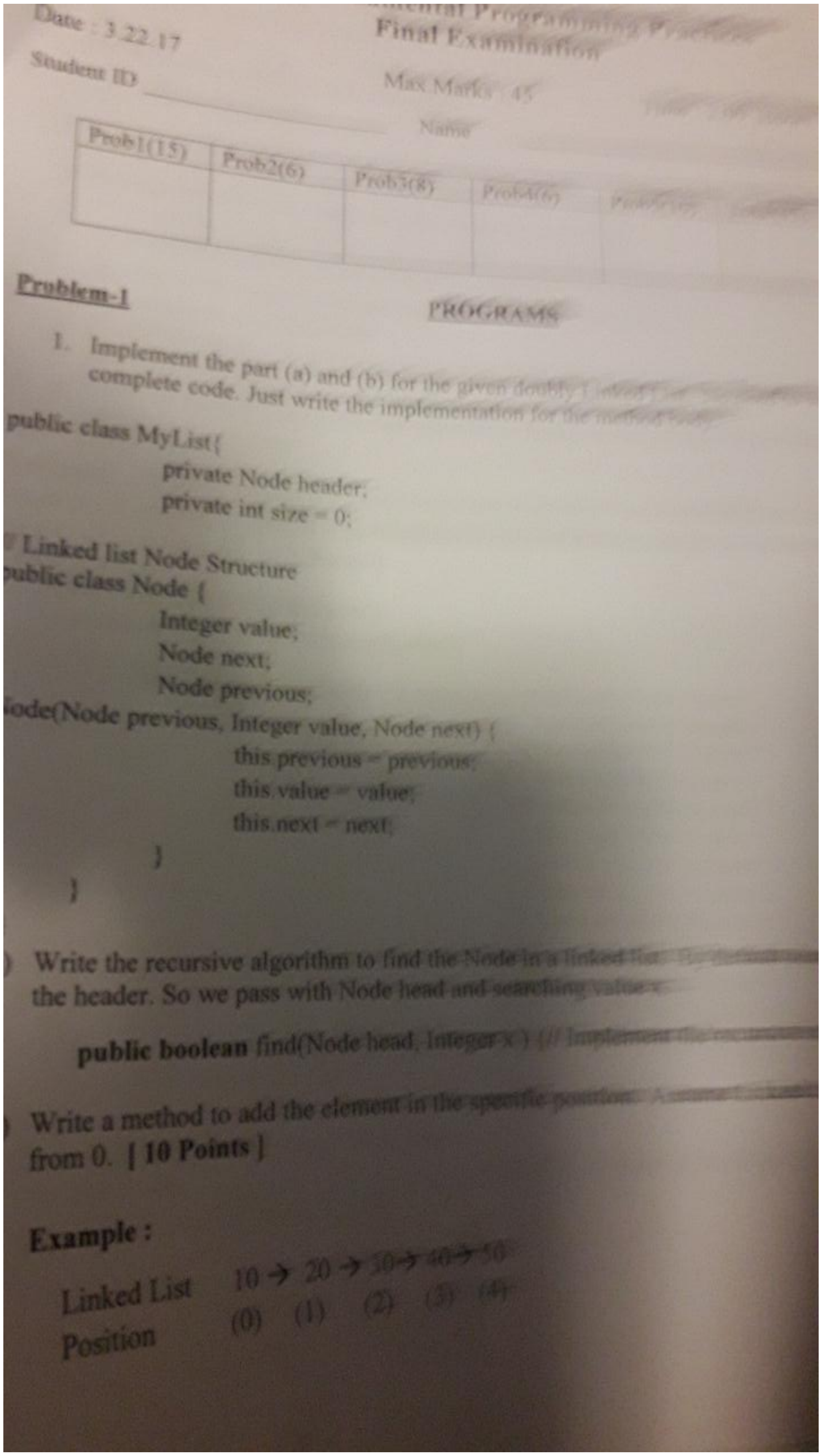
Hint from the API Map Interface hierarchy



Hint : To insert values in to the Map Collection use **put(Object Key, Object Va**
plit the word either use Arrays.Split(Char ch) or String.Split(Char ch),both retur

roblem-3 – Exception Handling **[8**

ewrite the given code by implementing two suitable exceptions from the API se exceptions.



Date : 3.22.17

Final Examination

Student ID ____________________

Max Marks : 45

Name

Prob1(15)	Prob2(6)	Prob3(8)	Prob4(6)	[illegible]

Problem-1

PROGRAMS

1. Implement the part (a) and (b) for the given doubly [illegible] complete code. Just write the implementation for the [illegible]

```
public class MyList{
                private Node header;
                private int size = 0;

// Linked list Node Structure
public class Node {
                Integer value;
                Node next;
                Node previous;
Node(Node previous, Integer value, Node next) {
                        this.previous = previous;
                        this.value = value;
                        this.next = next;
                }
    }
```

) Write the recursive algorithm to find the Node in a linked list [illegible] the header. So we pass with Node head and searching value x [illegible]

public boolean find(Node head, Integer x) {// Implement [illegible]

) Write a method to add the element in the specific position. [illegible] from 0. **[10 Points]**

Example :

Linked List	10 →	20 →	30 →	40 →	50
Position	(0)	(1)	(2)	(3)	(4)

```
        private double pop; // Population in Millions
        public String getCname() {
                return cname;
        }

        public void setCname(String cname) {
                this.cname = cname;
        }
        public double getPop() {
                return pop;
        }
        public void setPop(double pop) {
                this.pop = pop;
        }
        public TopCity(String cname, double pop) {
                this.cname = cname;
                this.pop = pop;
        }
        @Override
        public String toString() {
                String temp = String.format("%10.0f", this.pop);
                return  cname + " Population : " + temp + " million ";
        }
}
public class DataClass {
        static enum SortMethod {BYNAME, BYPOP};
        public final static List<TopCity> USACITY = new ArrayList<>();

        public static void main(String args[]){
        // Hint : 8_174_959 is double type value.(Java supports _ as a number sep
                USACITY.add(new TopCity("New York", 8_174_959));
                USACITY.add(new TopCity("Los Angeles", 3_792_657));
                USACITY.add(new TopCity("Chicago", 2_695_598));
                USACITY.add(new TopCity("Houston", 2_096_661));
                USACITY.add(new TopCity("Philadelphia", 1_526_006));
                USACITY.add(new TopCity("Phoenix", 1_447_617));
                USACITY.add(new TopCity("San Antonio", 1_327_556));
                sort(USACITY, SortMethod.BYNAME);
                sort(USACITY, SortMethod.BYPOP);
```

ic static void sort(List<TopCity> list, SortMethod type){

ite your implementation to sort the TopCity list based on the SortMethod

ır SortMethod type is BYNAME, then sort and print the list by using cit

r SortMethod type is BYPOP, then sort and print the list by using city

yntax of Comparator	**Hint : Syntax of sort fro**

PROGRAMS

Problem 1

1. Implement the part (a) and (b) for the given doubly Linked List. No
   complete code. Just write the implementation for the method body.

```
public class DLList{
        private Node header;
        private int size = 0;

// Linked list Node Structure
public class Node {
        Integer value;
        Node next;
        Node previous;
Node(Node previous, Integer value, Node next) {
                this.previous = previous;
                this.value = value;
                this.next = next;
        }
    }
}
```

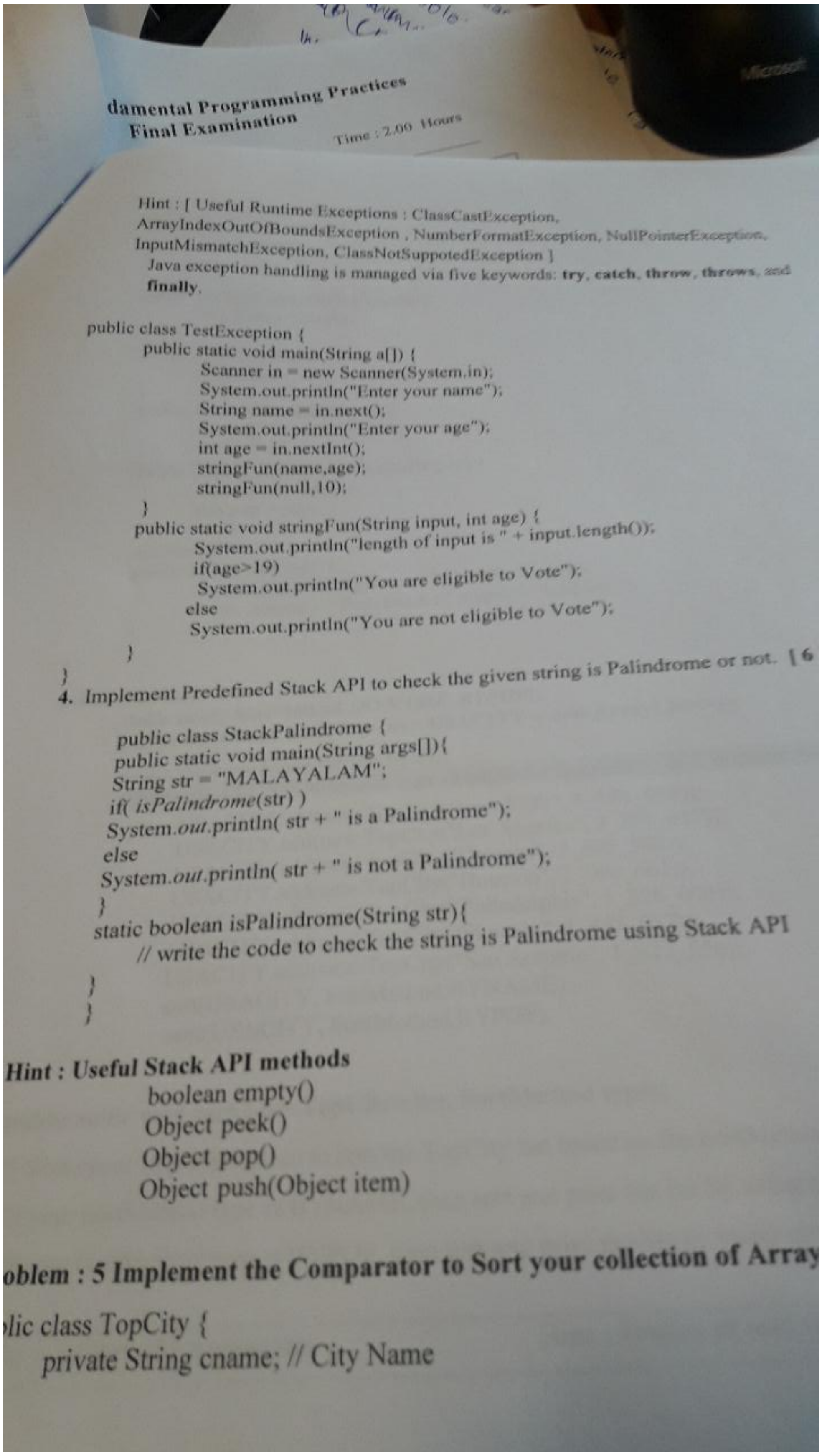
a) Write the recursive algorithm to find the Node in a linked list. By defau
   the header. So we pass with Node head and searching value x.

   public boolean find(Node head, Integer x) {// Implement the recur

b) Write a method to add the element in the specific position. Assume Lin
   [illegible] (10 Points)

   Example :

```
   Linked List      10 → 20 → 30 → 40 → 50
   Position         (0)   (1)   (2)   (3)   (4)
```



damental Programming Practices
Final Examination

Time : 2.00 Hours

Hint : [Useful Runtime Exceptions : ClassCastException, ArrayIndexOutOfBoundsException , NumberFormatException, NullPointerException, InputMismatchException, ClassNotSuppotedException]
Java exception handling is managed via five keywords: **try**, **catch**, **throw**, **throws**, and **finally**.

```
public class TestException {
        public static void main(String a[]) {
                Scanner in = new Scanner(System.in);
                System.out.println("Enter your name");
                String name = in.next();
                System.out.println("Enter your age");
                int age = in.nextInt();
                stringFun(name,age);
                stringFun(null,10);
        }
        public static void stringFun(String input, int age) {
                System.out.println("length of input is " + input.length());
                if(age>19)
                 System.out.println("You are eligible to Vote");
                else
                 System.out.println("You are not eligible to Vote");
        }
}
```

4. Implement Predefined Stack API to check the given string is Palindrome or not. [6

```
public class StackPalindrome {
public static void main(String args[]){
String str = "MALAYALAM";
if( isPalindrome(str) )
System.out.println( str + " is a Palindrome");
else
System.out.println( str + " is not a Palindrome");
}
static boolean isPalindrome(String str){
        // write the code to check the string is Palindrome using Stack API

}
}
```

Hint : Useful Stack API methods

```
        boolean empty()
        Object peek()
        Object pop()
        Object push(Object item)
```

oblem : 5 Implement the Comparator to Sort your collection of Array

```
lic class TopCity {
    private String cname; // City Name
```