IT22005

- Kth Smallest element

Code:

```
import java.util.*;
public class KthSmallestElement {
   public static int findKthSmallest(ArrayList<Integer> list, int k){
   Collections. sort (list);
   return list. get (k-1);
}
public static void main (string [ ] args){
  Scanner sc = new Scanner(system. in);
 ArrayList<Integer> list = new ArrayList <> ();
 int n = sc.nextInt();
for (int i=0; i<n; i++)
  list. add(sc. nextInt());
 int k = sc . nextInt();
 int smallest= findKthSmallest (list, k);
 System.out.println(" The "+k+" th smallest element
                          is : " + smallest);
  sc. close ();
 }
}
```

- Linked List Equality

Code:

```
 import java.util.*;
 public class Equality{
  public static void main( string [] args){
   LinkedList <Integer> list1 = new LinkedList<> (Arrays.asList(1,2,3));
LinkedList<Integer>list2 = new LinkedList<>(Arrays.asList (1,2,3));
LinkedList<Integer>list3 = new LinkedList<>(Arrays.asList (1,3,2));

System.out.println("list1 = =list2: " + list1. equals (list2));
System.out.println(" list1 == list 3 : " + list1. equals (list3));
}
}
```

- Mapping of Word Frequency ~~in~~using tree map .

Code:

```
import java.util.*;
public class WordFrequencyTreeMap{
  public static void main(string[] args){
    String text = " I am Maisha Haq . ";
    String [] words = text.split(" ");
TreeMap<String, Integer> map= new TreeMap<>();
for (String word: words){
 map.put(word, map.getOrDefault (word,0)+1);
 }
for(Map.Entry<String, Integer> entry: map.entrySet()){
System.out.println(entry.getKey()+"->"+entry.getValue());
}
}
}
```

- Stack and Queue using PriorityQueue with custom compator.

```
import java.util.*;
Class CustomElement {
    int value;
    int order;
  CustomElement (int value, int order){
    this.value=value;
    this.order=order;

 }
}
```

```
public class QueueStackUsingPriorityQueue{
    Static int counter=0;
public static void main(String[] args){
PriorityQueue<CustomElement>queue= new PriorityQueue
            <> (Comparator.comparingInt(a-> a.order));

PriorityQueue<CustomElement>stack=new PriorityQueue
        <> ((a,b)-> b.order - a.order);
for (int i: new int[] {10,20,30}){
 queue.add (new customElement(i, counter));
 stack.add (new CustomElement(i,counter));
 counter++; }
System.out.println ("Queue Order:");
while(! queue.isEmpty())
  System.out.println (queue.poll().value);
System.out.println("Stack Order:");
while (!stack.isEmpty())
  System.out.println(stack.poll().value);
}
}
```

- Student TreeMap

Code:

```
import java.util.*:
class Student{
    string name:
    int age;
Student(string name, int age){
    this.name =name;
    this.age =age:
}
public string toString(){
    return name + " ("+age + " years )"; } }
public class StudentTreeMap{
public static void main(String[ ] args){
TreeMap<Integer,Student>students =new TreeMap<>();
students.put(101,new Student("Maisha", 22));
Students.put (102, new Student("Mim," 22));
students.put (103, new student("Asa", 22));

for (Map.Entry<Integer,Student>entry:students.entrySet()){
  System.out.println("ID: "+entry.getKey()+"->"+entry.
                                              getValue());
}
}

}
```