

IMRAN HAMID

[D:IT-22010]

(1) find Kth smallest element in an arraylist.

Code:

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class KthElement {

    public static int KthSmallest (ArrayList<Integer> Array,
                                   int x)

    {
        if (x < 0 || x > Array.size()) {
            throw new IllegalArgumentException ("Invalid value of x");
        }
        else
        {
            Collections.sort (Array);
            return Array.get (x-1);
        }
    }

    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
```

```

ArrayList<Integer> Array = new ArrayList<>();
System.out.println("Enter number of elements");
int n = sc.nextInt();
System.out.println("Enter " + n + " elements");
for (int i = 0; i < n; i++) {
    Array.add(sc.nextInt());
}
System.out.println("Enter xth element to be searched");
int x = sc.nextInt();
try {
    int kthSmallest = KthSmallest(Array, x);
    System.out.println("The " + x + "th element is " + kthSmallest);
}
catch (IllegalArgumentException e) {
    System.out.println(e.getMessage());
}
}
}

```

(2) A TreeMap to store the mappings of words to their frequencies given text.

Code:

```
import java.util.Map;
import java.util.Scanner;
import java.util.TreeMap;

public class FreqWords {

    static void count_freq (String str) {
        Map <String, Integer> mp = new TreeMap <> ();
        String arr[] = str.split(" ");
        for (int i = 0; i < arr.length; i++) {
            if (mp.containsKey(arr[i])) {
                mp.put(arr[i], mp.get(arr[i]) + 1);
            }
            else {
                mp.put(arr[i], 1);
            }
        }
    }
}
```



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

```
{  
public static void main (String[] args) {
```

```
    Scanner sc = new Scanner(System.in);
```

```
    String str = sc.nextLine();
```

```
    count_freq(str);
```

```
}  
}
```

(3) A TreeMap to store the mapping of students IDs to their details.

Code

```
import java.util. Map;  
import java.util. Scanner;  
import java.util. TreeMap;  
  
public class mapIDNameOf  
public static void main (String[] args) {  
    TreeMap <Integer, String> studentMap = new TreeMap<>();  
    Scanner sc = new Scanner (System.in);  
  
    System.out.print ("Enter number of students");  
  
    int n = sc.nextInt();  
  
    sc.nextLine();  
  
    for (int i=0; i<n; i++) {  
        System.out.print ("Enter student ID ");  
  
        int id = sc.nextInt();  
  
        sc.nextLine();
```

```

System.out.print("Enter student details ");
String details = sc.nextLine();
studentMap.put(id, details);
}
System.out.println("\n -- Student Details (sorted by ID) -- ");
for (Map.Entry<Integer, String> entry : studentMap.entrySet())
{
    System.out.print("ID : " + entry.getKey() + " -> Details: "
        + entry.getValue());
}
}
}
}

```

(4) A program to check if two linked lists are equal.

Code:

```
class Node {
```

```
    int data;
```

```
    Node next;
```

```
    Node (int data) {
```

```
        this.data = data;
```

```
        this.next = null;
```

```
class IdenticalLinkedList {
```

```
    static boolean areIdentical (Node head1, Node head2)
```

```
    {  
        while (head1 != null && head2 != null) {
```

```
            if (head1.data != head2.data)
```

```
                return false;
```

```
            head1 = head1.next;
```

```
            head2 = head2.next;
```

```
        }
```

```
        return (head1 == null && head2 == null);
```

9



```

public static void main (String[] args) {
    Node head1 = new Node(3);
    head1.next = new Node(2);
    head1.next.next = new Node(1);

    Node head2 = new Node(1);
    head2.next = new Node(2);
    head2.next.next = new Node(3);

    if (areIdentical(head1, head2) == true)
        System.out.println("Equal");
    else
        System.out.print("Not equal");
}
}

```



(5) A hashmap to store the mapping of employees IDs to their dept.

Code:

```
import java.util. HashMap;  
import java.util. Map;  
import java.util. Scanner;  
  
public class HashMapIdDept {  
    public static void main (String [] args) {  
        HashMap <Integer, String> employeeMap = new HashMap <> ();  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter number of employees");  
        int n = sc.nextInt();  
        sc.nextLine();  
        for (int i = 0; i < n; i++) {  
            System.out.println ("Enter employee ID (integer): ");  
            int id = sc.nextInt();  
            sc.nextLine();
```

```
for (int i=0; i<n; i++)
```

```
System.out.println("Enter dept : ");
```

```
String dept = sc.nextLine();
```

```
employeeMap.put(id, dept);
```

```
{
```

```
System.out.println("Employee Details ");
```

```
for (Map.Entry<Integer, String> entry : employeeMap.entrySet())
```

```
{
```

```
System.out.println("Employee ID : " + entry.getKey() + " →
```

```
Department : " + entry.getValue());
```

```
{
```

```
{
```

```
}
```