Rudro

IT-22002

Program-1: Find kth smallest element in an Array List.

Soln:

```
// importing required classes from Java standard library.
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class kthSmallest {
    // Method to find kth smallest element from ArrayList
    public static int KthSmallest(ArrayList<Integer>
                                  Array, int n)
    {
        // Check for invalid input
        if(n<0 || n>Array.size())
        {
            // Throw exception with a message
            throw new IllegalArgumentException
                ("Invalid value of n.");
        }
        else {
```

```
        Collections.sort(Array); //Sort ArrayList in
                                    ascending order
        return Array.get(n-1); // Return the (n-1)th
    }                           index value
}

public static void main(String[] args) {
    // Create Scanner object to read input
    Scanner sc = new Scanner(System.in);

    ArrayList<Integer> Array = new ArrayList();

    System.out.println("Enter the number of
                        elements:");
    int n = sc.nextInt(); // Read number of elements.
    System.out.println("Enter " + n + " elements:");
    for(int i = 0; i<n; i++)
    {
        Array.add(sc.nextInt());
    }

    System.out.println("Enter the nth element
                        to be searched:"); // Prompt user to
                                            enter value of n
    int n = sc.nextInt();
```

```
try{
    int kthsmallest = kthSmallest(Array, n); // Call function
    System.out.println("The "+n+"th smallest element is: "+kthsmallest); // Print result
}
catch(IllegalArgumentException e){
    System.out.println(e.getMessage()); // Display error message if exception occurs.
}
}
}
```

Output:

```
Enter the number of elements:
5
Enter 5 elements:
10 9 8 5 6
Enter the nth element to be searched:
4
The 4th smallest element is 9.
```

Program-2: TreeMap program to store the mappings of words to their frequencies in a given text.

Soln:

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

public class FreqWords{
    // Function to count frequency of words
       in given text.
    static void count_freq(String str)
    {
        Map<String, Integer> mp = new TreeMap
                                  <>();
        // Splitting to find word
        String arr[] = str.split(" ");
        // Loop to iterate over words.
```

```
for(int i=0;i<arr.length;i++)
{
    // Conditions to check if array element is
       present in the hash-map
    if(mp.containsKey(arr[i]))
    {
        mp.put(arr[i],mp.get(arr[i])+1);
    }
    else{
        mp.put(arr[i],1);
    }
}
//Loop to iterate over elements of map.
for(Map.Entry<String,Integer>entry:
        mp.entrySet())
{
    System.out.println(entry.getKey()+
        "-"+entry.getValue());
}
```

```
// Entry point of the program:
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        String str = sc.nextLine();
        // Function call
        count_freq(str);
    }
}
```

Output:

Software Engineering is the best passion in the world

Engineering-1

Software-1

best-1

in-1

is-1

passion-1

the-2

world-1

Program-3: Treemap program to store the mappings of student IDs to their details.

Soln:

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

public class mapIDName {
    public static void main (String[] args) {
        //TreeMap stores key in sorted order
        TreeMap<Integer, String> studentMap=
                new TreeMap<>();
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number
                of students:");
        int n=sc.nextInt();
        sc.nextLine(); // Consume newline.
```

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

    System.out.print("Enter student ID
                    (integer): ");

    int id = sc.nextInt();

    sc.nextLine();

    System.out.print("Enter student details
                    (e.g. Name, Department): ");

    String details = sc.nextLine();

    studentMap.put(id, details);

}

//Displaying student data is sorted order of
    IDs

System.out.println("\n--- Student Details
                    (Sorted by ID) ---");

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

```
System.out.println("ID: " + entry.getKey() + " → Details: " + entry.getValue());
        }
    }
}
```

Output:

```
Enter the number of students: 2
Enter student ID (integer): 1
Enter student details (e.g., Name, Department):
Atif, ICT
Enter student ID (integer): 2
Enter student details (e.g., Name, Department):
Ratul, ICT

--- Student Details (Sorted by ID) ---
ID: 1 → Details: Atif, ICT
ID: 2 → Details: Ratul, ICT
```

Program-4: A program to check if two Linked lists are equal.

Soln:

```
class Node{
    int data;
    Node next;
    Node(int data)
    {
        this.data=data;
        this.next=null;
    }
}

class IdenticalLinkedList{
    // Returns true if two linked lists are equal
    static boolean areIdentical(Node head1,
                                Node head2)
    {
        while(head1!=null && head2!=null){
            if(head1.data!=head2.data)
                return false;
```

```
// Move to next nodes in both list
head1 = head1.next;
head2 = head2.next;
}
// If linked lists are identical, both head1 and
head2 must be null.
return (head1 == null && head2 == null);
}
public static void main(String[] args) {
// Create first linked list: 3 → 2 → 1
Node head1 = new Node(3);
head1.next = new Node(2);
head1.next.next = new Node(1);
// Create second linked list: 1 → 2 → 3
Node head2 = new Node(1);
head2.next = new Node(2);
head2.next.next = new Node(3);
```

```
// Function call
if (areIdentical(head1, head2) == true)
{
    System.out.println("The linked lists are
        identical.");
}
else {
    System.out.println("The linked lists are not
        identical.");
}
}
```

Output:

The linked lists are not identical.

Program-5: A program to store mappings of employee Ids to their departments.

Soln:

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

public class hashMapIDDept {
    public static void main(String[] args) {
        // HashMap to store employee ID →
            Depantment

        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(); // Consume newline.
        for(int i=0; i<n; i++) {
```

```
System.out.println("Enter employee ID (integer
                    ");
    int id = sc.nextInt();
    sc.nextLine(); // Consume newline
    System.out.println("Enter department: ");
    String department = sc.nextLine();
    employeeMap.put(id, department);
}

// Display employee ID and their department
                                    Department
System.out.println("\n--- Employee Dep
                    Mappings ---");

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

```
    }
  }
}
```

Output:

```
Enter number of employees:
1
Enter employee ID (integer):
1000
Enter department:
Human Resources

--- Employee Department Mappings ---
Employee ID: 1000 → Department: Human Resources
```