

Q) smallest

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

public class smallest {
    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 the 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 the xth element to be  
searched: ");
```

```
    int x = sc.nextInt();
```

```
    try {
```

```
        int kthsmallest = kthsmallest(Array, x);
```

```
        System.out.println("The " + x + "th smallest element  
+ kthsmallest);
```

```
    }
```

```

catch (IllegalArgumentException e)
{
    System.out.println (e.getMessage());
}
}
}

```

(b) frequency words

```

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

public class frequency {
    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();  
    countFreq (str);  
}  
}
```

Q. Write a Java program to implement a Map. (5 marks)

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

public class Mapid {
    public static void main(String [] args) {
        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();

        for(int i=0; i<n; i++) {
            System.out.println("Enter student ID (Integer): ");
            int id = sc.nextInt();
            sc.nextLine();
            System.out.println("Enter student details (e.g., name, dept): ");
            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.println("ID: " + entry.getKey() + " -> Details: "
        + entry.getValue());
}
}
}
}

```

⑤ Linked List

```

class Node {
    int data;
    Node next;

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

```

```
class linked_list {
```

```
    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);
```

```
    }
```

```
    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("The Linked Lists are identical.");
else
    System.out.println("They are not identical.");
}
}

```

⑥ HashMap

```

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

public class hashmap {
    public static void main (String [] args) {
        HashMap <Integer, String> employeeMap = new HashMap<>();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the 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();
    System.out.println("Enter department: ");
    String department = sc.nextLine();
    employeeMap.put(id, department);
}

```

```

System.out.println("\n--- Employee-dept mapping ---");

```

```

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

```