

ID- IT22052

Mim Akter Mou

1) k-th Smallest Element

```
import java.util.Arrays;  
public class kthSmallestElement{  
    public static int findkthSmallest(int[] arr, int k){  
        Arrays.sort(arr); // Sort the array  
        return arr[k-1]; // Return the k-th smallest  
        // element (1-based index)  
    }  
    public static void main(String[] args){  
        int[] array = {7, 15, 2, 9, 8, 3, 20};  
        int k = 4;  
        int result = findkthSmallest(array, k);  
        System.out.println(k + "nd smallest element is:" +  
        result);  
    }  
}
```

2] WordFrequency Map

```

import java.util.*;
public class WordFreq {
    public static void main(String[] args) {
        String text = "hello ICT";
        Map<String, Integer> map = new HashMap<>();
        for (String word : text.toLowerCase().split(" \w+")) {
            map.put(word, map.getOrDefault(word, 0) + 1);
        }
        System.out.println(map);
    }
}

```

3] Queue Stack Using Priority Queue

```

import java.util.*;
public class QueueStackWithPriorityQueue {
    static class Element {
        int value, order;
        Element(int value, int order) {
            this.value = value;
            this.order = order;
        }
    }
}

```

ID-IT 22052

Mim Akten Notizen

```
public static void main(String[] args) {
    int orden = 0;
    Priority Queue<Element> queue = new Priority Queue<
        Comparator<Element>() {
            public int compare(Element e1, Element e2) {
                return e1.getWert() - e2.getWert();
            }
        }
    };
    queue.offer(new Element(10, orden++));
    queue.offer(new Element(30, orden++));
    queue.offer(new Element(50, orden++));
    System.out.println("Queue (FIFO):");
    while (!queue.isEmpty()) {
        System.out.println(queue.poll().value);
    }
}
```

1
orden = 0;

```
Priority Queue<Element> stack = new Priority Queue<>((a, b) {
    return b.orden - a.orden;
});
stack.offer(new Element(10, orden++));
stack.offer(new Element(30, orden++));
stack.offer(new Element(50, orden++));
System.out.println("Stack (LIFO):");
while (!stack.isEmpty()) {
    System.out.println(stack.poll().value);
}
}
```

3 3 }

ID-IT22052

Mim Afsen Mon

4 Student Tree Map

```
import java.util.*;  
public static class StudentDetails{  
    public static void main(String[] args){  
        TreeMap<Integer, String> students = new TreeMap<>();  
        students.put(102, "Shamima, 92");  
        students.put(101, "Muskan, 90");  
        students.put(103, "Mahi, 93");  
        for(Entry<Integer, String> e : students.entrySet())  
            System.out.println("ID :" + e.getKey() + ", Details :" +  
                e.getValue());  
    }  
}
```

5 Compare Two Linked Lists

```
import java.util.*;  
public class LinkedListEqual {  
    public static void main(String[] args){  
        LinkedList<Integer> list1 = new LinkedList<>(  
            List.of(1, 2, 3));  
    }  
}
```

ID-IT22052

Mim Akten Mon

```
LinkedList<Integer> list2 = new LinkedList<Integer>();
(1, 2, 3));
if (list1.equals(list2))
    System.out.println("Lists are equal");
else
    System.out.println("Lists are not equal");
}
}
```

⑥ Employee ID to Department using HashMap

```
import java.util.*;
public class EmployeeMap
public static void main (String [] args)
    HashMap<Integer, String> empMap = new HashMap();
    empMap.put (101, "CSE");
    empMap.put (102, "BGE");
    empMap.put (103, "ICT");
    for (var e: empMap.entrySet())
        System.out.println ("ID" + e.getKey () + " Dept: " +
            e.getValue ());
}
}
```