

* Linked List Equality Check

```
import java.util.LinkedList;

public class LinkedListEqualityCheck {

    public static boolean areEqual(LinkedList<Integer>
        list1, LinkedList<Integer> list2) {
        if (list1 == null || list2 == null) return false;
        if (list1.size() != list2.size()) {
            return false;
        }
        for (int i = 0; i < list1.size(); i++) {
            if (!list1.get(i).equals(list2.get(i)))
                return false;
        }
        return true;
    }
}
```

```
public static void main(String[] args) {
    LinkedList<Integer> list1 = new LinkedList<>();
    LinkedList<Integer> list2 = new LinkedList<>();
}
```


17-22020

list1.add(10);

list1.add(20);

list1.add(30);

list2.add(10);

list2.add(20);

list2.add(40);

if (areEqual(list1, list2))

System.out.println("The linked lists are equal.");

else

System.out.println("The linked lists are not equal.");

}

}

Word frequency using map

import java.util.Map;

import java.util.Scanner;

import java.util.TreeMap;

public class WordFrequency-Using-Maps

17-22020

public static void main (String args[]) {

Scanner scanner = new Scanner (System.in);

System.out.print ("Enter a line of text : ");

String input = scanner.nextLine();

String[] words = input.toLowerCase().split("\\W+");

TreeMap<String, Integer> WordCount = new TreeMap<>();

for (String word : words) {

if (word.isEmpty()) continue;

WordCount.put (word, WordCount.getOrDefault (

word, 0) + 1);

System.out.println ("Word Frequencies

(sorted) : ");


```

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

```

```

    scanner.close();
}
}

```

Employee and their department using map

```

import java.util.HashMap;

```

```

import java.util.Map;

```

```

import java.util.Scanner;

```

```

public class EmployeeDepartmentMap {

```

```

    public static void main (String[] args) {

```

```

        Scanner scanner = new Scanner (System.in);

```

```

        HashMap (Integer, String) employeeMap = new
            HashMap < > ();

```



```
System.out.print("Enter the number of employee:");
```

```
int n = scanner.nextInt();
```

```
scanner.nextLine();
```

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

```
{  
    System.out.println("\nEnter details for employee #
```

```
(i+1) + ":");  
    System.out.print("Employee ID (Integer): ");
```

```
int empld = scanner.nextInt();
```

```
scanner.nextLine();
```

```
System.out.print("Department: ");
```

```
String department = scanner.nextLine();
```

```
employeeMap.put(empld, department);
```

```
}  
System.out.println("\n Employee ID to department
```

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

```
{  
    System.out.print("Employee ID : " + entry.getKey()
```

```
    + " -> Department : " + entry.getValue());  
}
```


Kth-Smallest-Element-Arraylist

```
import java.util.*;  
import java.util.Collections;  
public class Kth-Smallest-Element-Arraylist {  
    public static int findKthSmallest(  
        ArrayList<Integer> list, int k) {  
        if (k <= 0 || k > list.size())  
            throw new IllegalArgumentException("Invalid value of k : " + k);  
        Collections.sort(list);  
        return list.get(k-1);  
    }  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        ArrayList<Integer> list = new ArrayList<>();  
        System.out.print("\nEnter the number of element : ");  
        int n = scanner.nextInt();  
    }  
}
```

```
System.out.println("Enter the element: ");
```

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

```
list.add(scanner.nextInt());
```

```
System.out.print("Enter k: ");
```

```
int k = scanner.nextInt();
```

```
try {
```

```
int kthSmallest = findKthSmallest(list, k);
```

```
System.out.println("The " + k + "th smallest
```

```
element is: " + kthSmallest);
```

```
} catch (IllegalArgumentException e) {
```

```
System.out.println(e.getMessage());
```

```
scanner.close();
```

```
}
```


Student ID and details using map

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

class Student {
    String name;
    int age;
    String department;

    Student(String name, int age, String department) {
        this.name = name;
        this.age = age;
        this.department = department;
    }

    public String toString() {
        return "Name " + name + ", Age: " + age +
            ", Department: " + department;
    }
}

public class StudentID-And-Details-Map {
```



```

public static void main (String[] args) {
    TreeMap< Integer, student > studentMap = new
        TreeMap<>();
    Scanner scanner = new Scanner(System.in);
    System.out.print ("Enter the number of
        student :");
    int n = scanner.nextInt();
    for (int i=0; i<n; i++) {
        System.out.println ("Enter details for
            student " + (i+1) + " :");
        System.out.print ("Student ID (integer):");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print ("Name: ");
        String name = scanner.nextLine();
        System.out.print ("Age: ");
        int age = scanner.nextInt();
        scanner.nextLine();
    }
}

```



```
System.out.print("Department: ");
```

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

```
Student student = new Student(name, age, dept);
```

```
student.put(id, student);
```

```
}
```

```
System.out.println("Student Records (sorted by
```

```
ID):");
```

```
for (Map.Entry<Integer, Student> entry :
```

```
studentMap.entrySet())
```

```
System.out.print("ID" + entry.getKey() + " -> " +
```

```
entry.getValue());
```

```
scanner.close();
```

```
}  
}
```