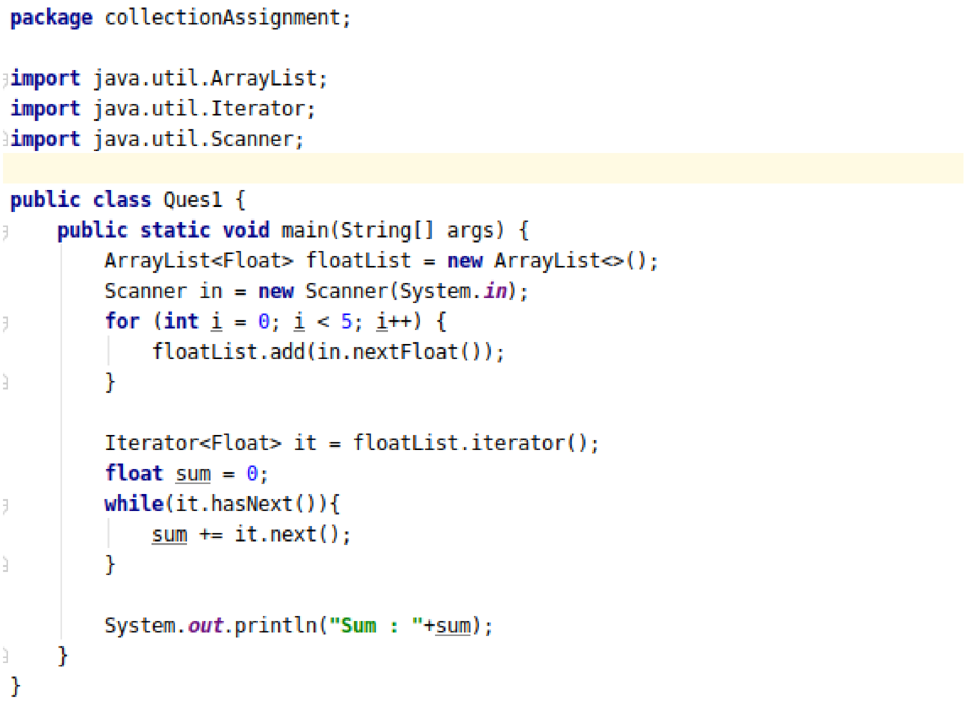
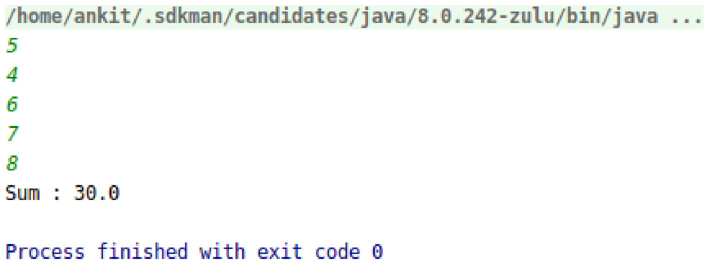
Question: 1:

Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.



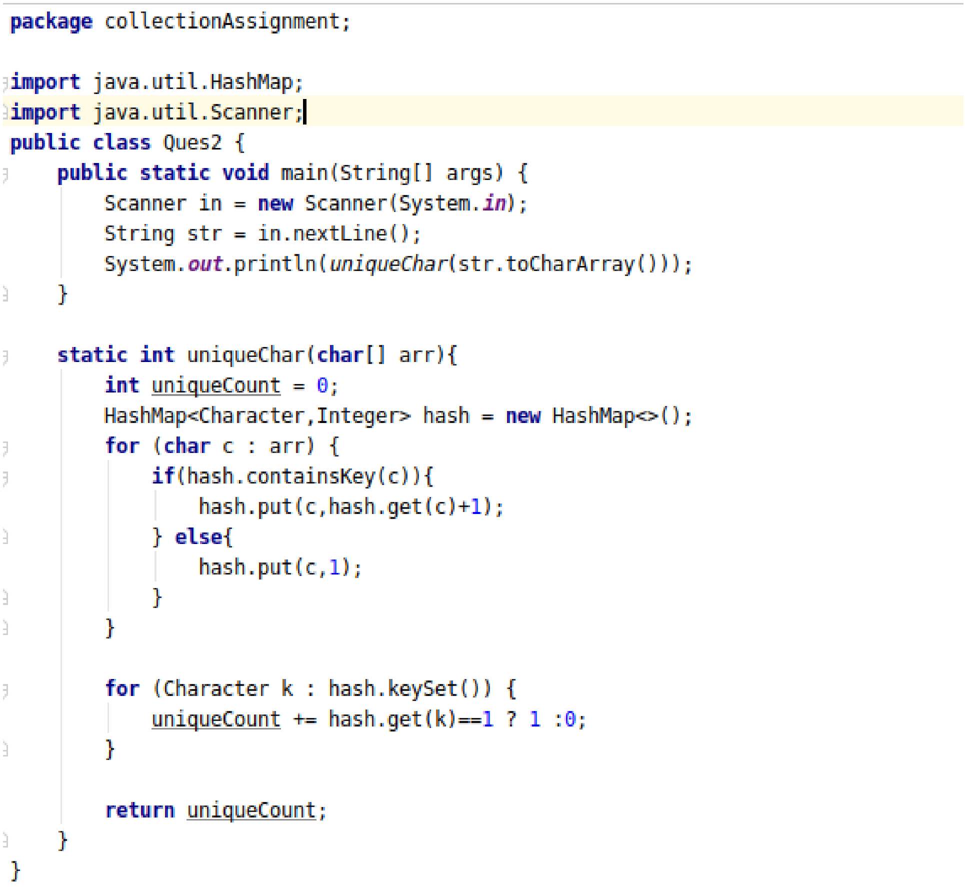
```java
package collectionAssignment;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

public class Ques1 {
    public static void main(String[] args) {
        ArrayList<Float> floatList = new ArrayList<>();
        Scanner in = new Scanner(System.in);
        for (int i = 0; i < 5; i++) {
            floatList.add(in.nextFloat());
        }

        Iterator<Float> it = floatList.iterator();
        float sum = 0;
        while(it.hasNext()){
            sum += it.next();
        }

        System.out.println("Sum : "+sum);
    }
}
```



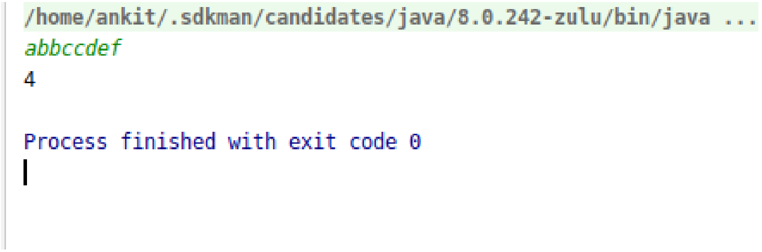
```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
5
4
6
7
8
Sum : 30.0

Process finished with exit code 0
```

Question: 2:

Write a method that takes a string and returns the number of unique characters in the string.

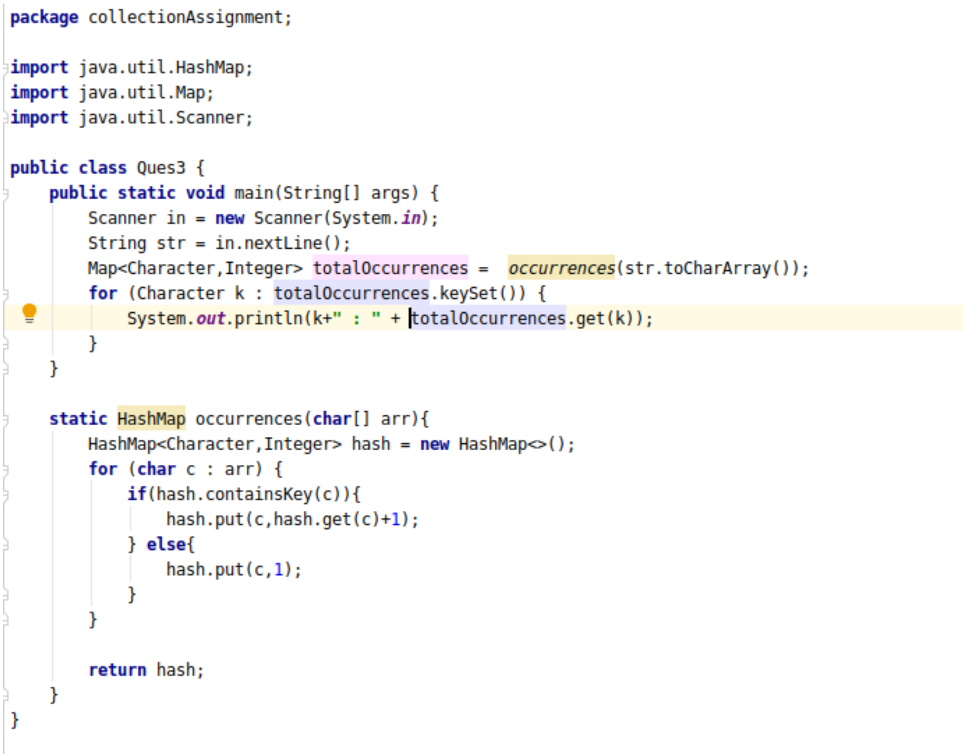


```java
package collectionAssignment;

import java.util.HashMap;
import java.util.Scanner;
public class Ques2 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        System.out.println(uniqueChar(str.toCharArray()));
    }

    static int uniqueChar(char[] arr){
        int uniqueCount = 0;
        HashMap<Character,Integer> hash = new HashMap<>();
        for (char c : arr) {
            if(hash.containsKey(c)){
                hash.put(c,hash.get(c)+1);
            } else{
                hash.put(c,1);
            }
        }

        for (Character k : hash.keySet()) {
            uniqueCount += hash.get(k)==1 ? 1 :0;
        }

        return uniqueCount;
    }
}
```

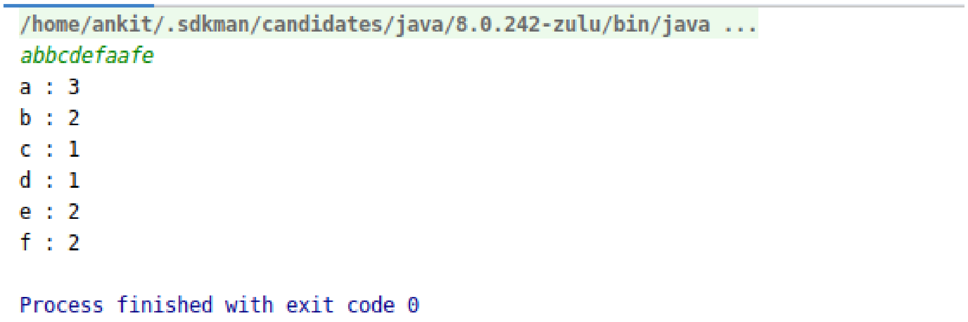


```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
abbccdef
4

Process finished with exit code 0
```

Question: 3:
Write a method that takes a string and print the number of occurrence of each character characters in the string.

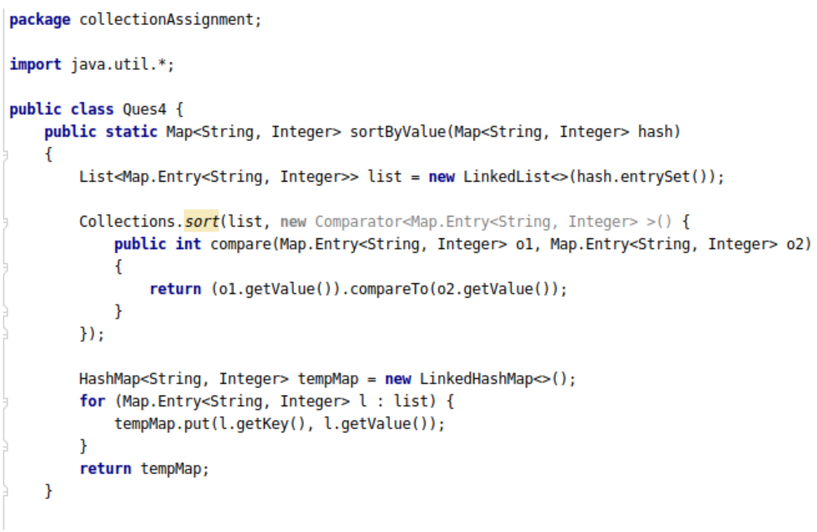


```java
package collectionAssignment;

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

public class Ques3 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        Map<Character,Integer> totalOccurrences =  occurrences(str.toCharArray());
        for (Character k : totalOccurrences.keySet()) {
            System.out.println(k+" : " + totalOccurrences.get(k));
        }
    }

    static HashMap occurrences(char[] arr){
        HashMap<Character,Integer> hash = new HashMap<>();
        for (char c : arr) {
            if(hash.containsKey(c)){
                hash.put(c,hash.get(c)+1);
            } else{
                hash.put(c,1);
            }
        }

        return hash;
    }
}
```

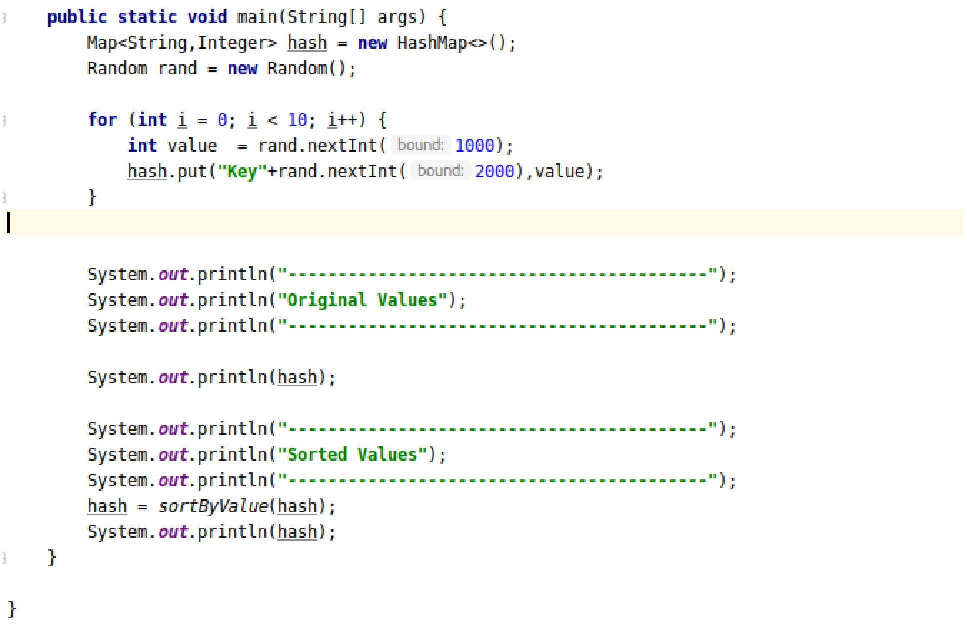


```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
abbcdefaafe
a : 3
b : 2
c : 1
d : 1
e : 2
f : 2

Process finished with exit code 0
```

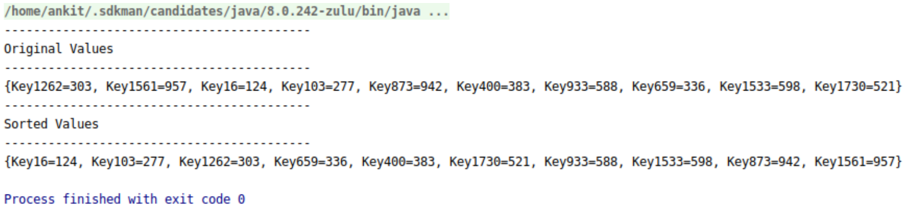
Question: 4:
Write a program to sort HashMap by value.



```java
package collectionAssignment;

import java.util.*;

public class Ques4 {
    public static Map<String, Integer> sortByValue(Map<String, Integer> hash)
    {
        List<Map.Entry<String, Integer>> list = new LinkedList<>(hash.entrySet());

        Collections.sort(list, new Comparator<Map.Entry<String, Integer> >() {
            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2)
            {
                return (o1.getValue()).compareTo(o2.getValue());
            }
        });

        HashMap<String, Integer> tempMap = new LinkedHashMap<>();
        for (Map.Entry<String, Integer> l : list) {
            tempMap.put(l.getKey(), l.getValue());
        }
        return tempMap;
    }
```

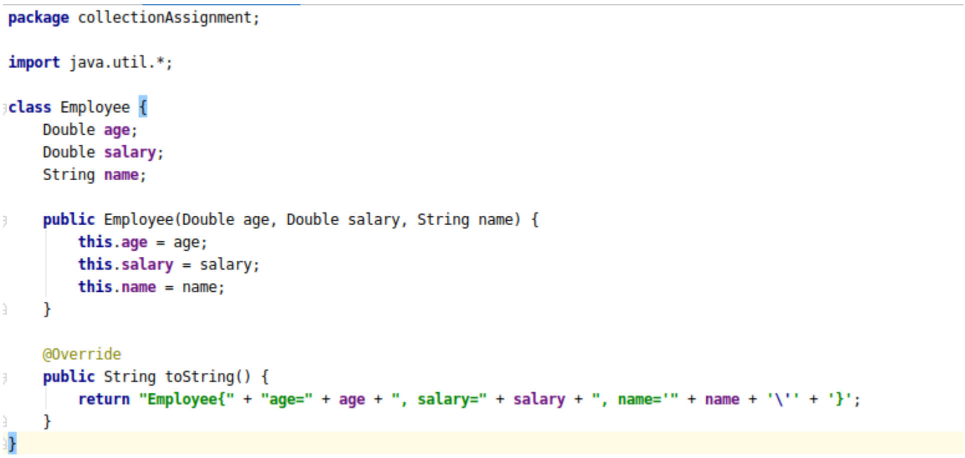


```java
    public static void main(String[] args) {
        Map<String,Integer> hash = new HashMap<>();
        Random rand = new Random();

        for (int i = 0; i < 10; i++) {
            int value  = rand.nextInt( bound: 1000);
            hash.put("Key"+rand.nextInt( bound: 2000),value);
        }


        System.out.println("------------------------------------------");
        System.out.println("Original Values");
        System.out.println("------------------------------------------");

        System.out.println(hash);

        System.out.println("------------------------------------------");
        System.out.println("Sorted Values");
        System.out.println("------------------------------------------");
        hash = sortByValue(hash);
        System.out.println(hash);
    }

}
```

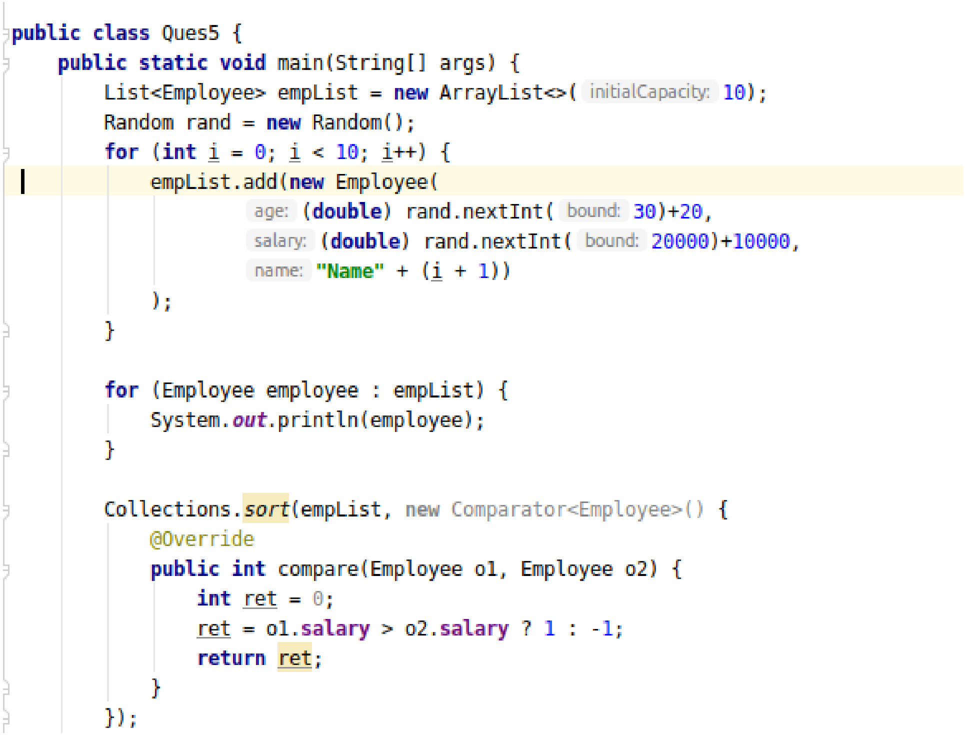


```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
------------------------------------------
Original Values
------------------------------------------
{Key1262=303, Key1561=957, Key16=124, Key103=277, Key873=942, Key400=383, Key933=588, Key659=336, Key1533=598, Key1730=521}
------------------------------------------
Sorted Values
------------------------------------------
{Key16=124, Key103=277, Key1262=303, Key659=336, Key400=383, Key1730=521, Key933=588, Key1533=598, Key873=942, Key1561=957}

Process finished with exit code 0
```

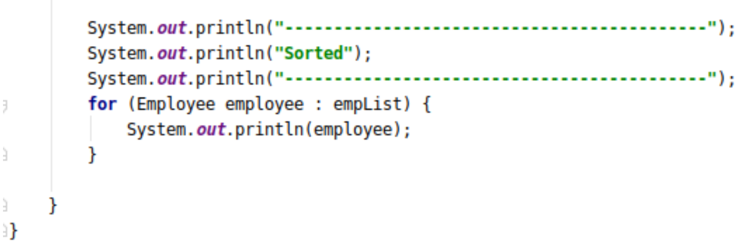
Question: 5:
Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name



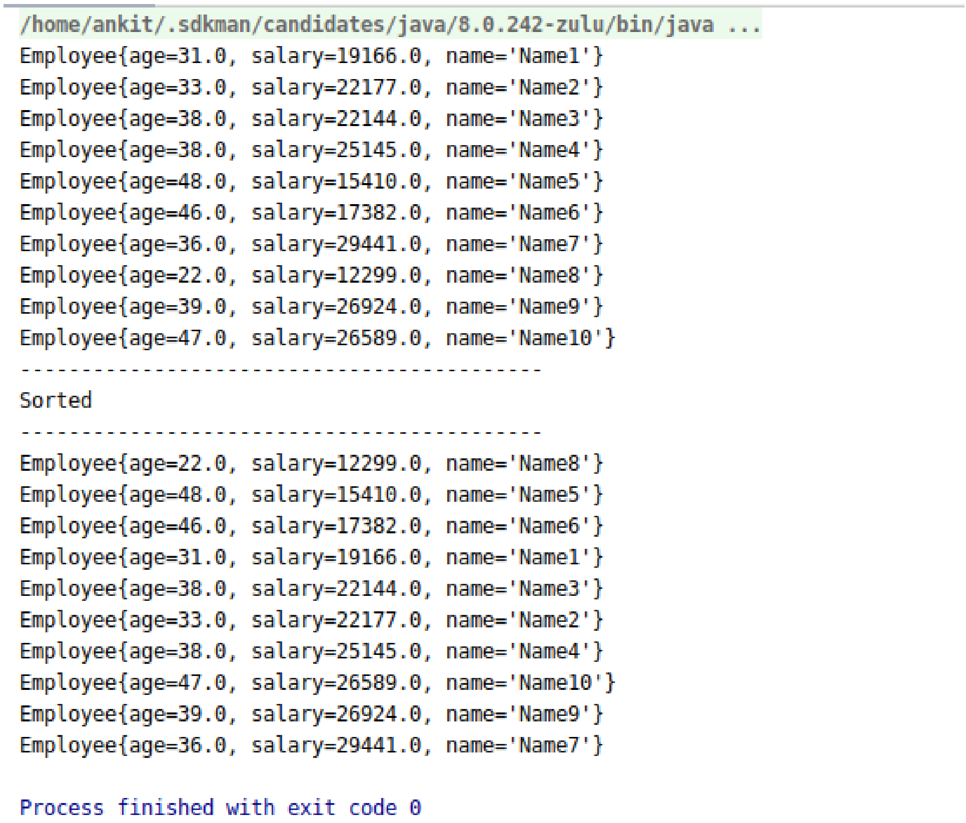
```
package collectionAssignment;

import java.util.*;

class Employee {
    Double age;
    Double salary;
    String name;

    public Employee(Double age, Double salary, String name) {
        this.age = age;
        this.salary = salary;
        this.name = name;
    }

    @Override
    public String toString() {
        return "Employee{" + "age=" + age + ", salary=" + salary + ", name='" + name + '\'' + '}';
    }
}
```



```
public class Ques5 {
    public static void main(String[] args) {
        List<Employee> empList = new ArrayList<>( initialCapacity: 10);
        Random rand = new Random();
        for (int i = 0; i < 10; i++) {
            empList.add(new Employee(
                    age: (double) rand.nextInt( bound: 30)+20,
                    salary: (double) rand.nextInt( bound: 20000)+10000,
                    name: "Name" + (i + 1))
            );
        }

        for (Employee employee : empList) {
            System.out.println(employee);
        }

        Collections.sort(empList, new Comparator<Employee>() {
            @Override
            public int compare(Employee o1, Employee o2) {
                int ret = 0;
                ret = o1.salary > o2.salary ? 1 : -1;
                return ret;
            }
        });
```



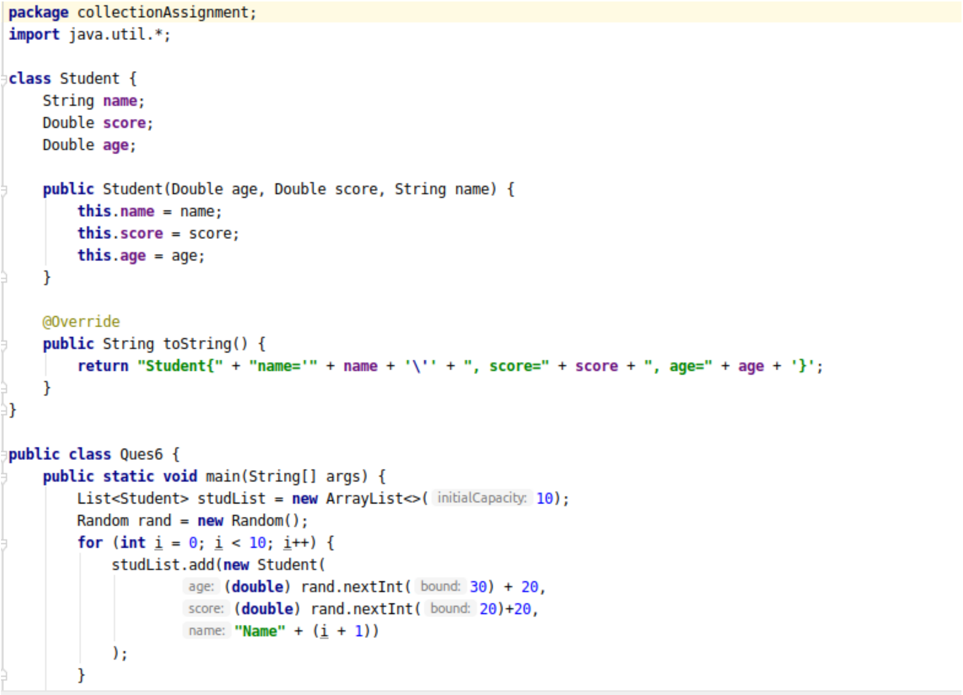
```
        System.out.println("-------------------------------------------");
        System.out.println("Sorted");
        System.out.println("-------------------------------------------");
        for (Employee employee : empList) {
            System.out.println(employee);
        }

    }
}
```



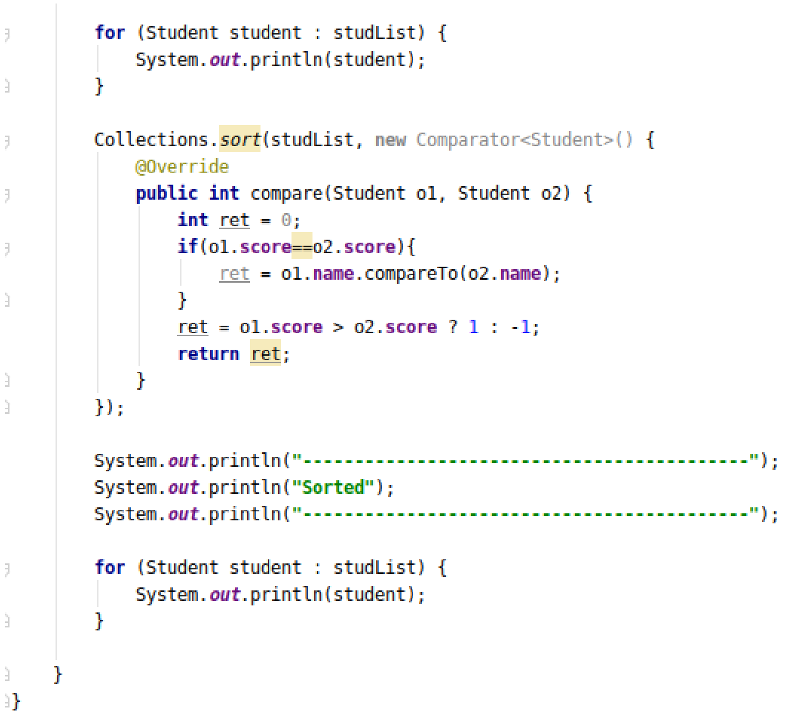
```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Employee{age=31.0, salary=19166.0, name='Name1'}
Employee{age=33.0, salary=22177.0, name='Name2'}
Employee{age=38.0, salary=22144.0, name='Name3'}
Employee{age=38.0, salary=25145.0, name='Name4'}
Employee{age=48.0, salary=15410.0, name='Name5'}
Employee{age=46.0, salary=17382.0, name='Name6'}
Employee{age=36.0, salary=29441.0, name='Name7'}
Employee{age=22.0, salary=12299.0, name='Name8'}
Employee{age=39.0, salary=26924.0, name='Name9'}
Employee{age=47.0, salary=26589.0, name='Name10'}
-------------------------------------------
Sorted
-------------------------------------------
Employee{age=22.0, salary=12299.0, name='Name8'}
Employee{age=48.0, salary=15410.0, name='Name5'}
Employee{age=46.0, salary=17382.0, name='Name6'}
Employee{age=31.0, salary=19166.0, name='Name1'}
Employee{age=38.0, salary=22144.0, name='Name3'}
Employee{age=33.0, salary=22177.0, name='Name2'}
Employee{age=38.0, salary=25145.0, name='Name4'}
Employee{age=47.0, salary=26589.0, name='Name10'}
Employee{age=39.0, salary=26924.0, name='Name9'}
Employee{age=36.0, salary=29441.0, name='Name7'}

Process finished with exit code 0
```

Question: 6:

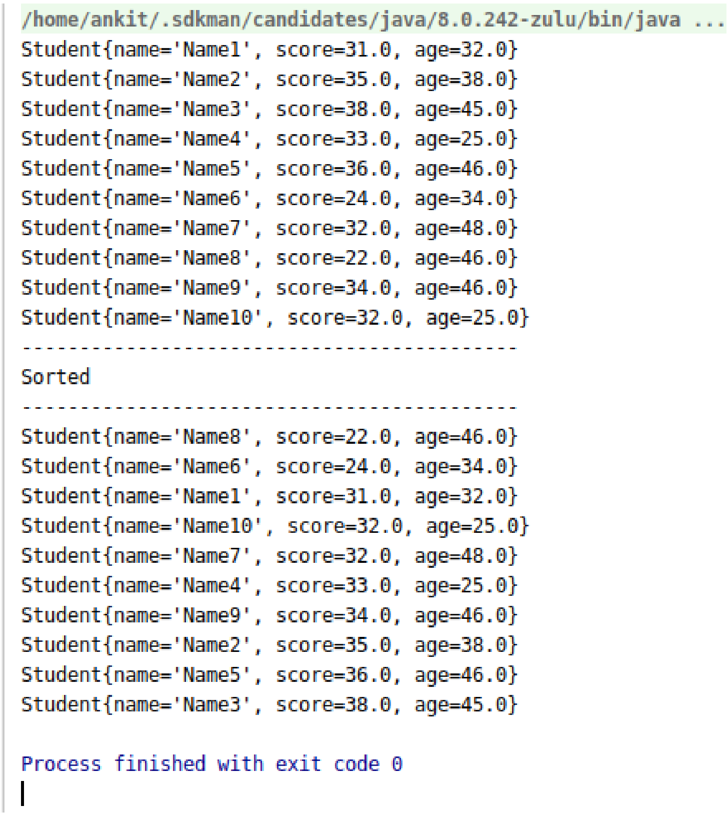
Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age



```java
package collectionAssignment;
import java.util.*;

class Student {
    String name;
    Double score;
    Double age;

    public Student(Double age, Double score, String name) {
        this.name = name;
        this.score = score;
        this.age = age;
    }

    @Override
    public String toString() {
        return "Student{" + "name='" + name + '\'' + ", score=" + score + ", age=" + age + '}';
    }
}

public class Ques6 {
    public static void main(String[] args) {
        List<Student> studList = new ArrayList<>( initialCapacity: 10);
        Random rand = new Random();
        for (int i = 0; i < 10; i++) {
            studList.add(new Student(
                    age: (double) rand.nextInt( bound: 30) + 20,
                    score: (double) rand.nextInt( bound: 20)+20,
                    name: "Name" + (i + 1))
            );
        }
```



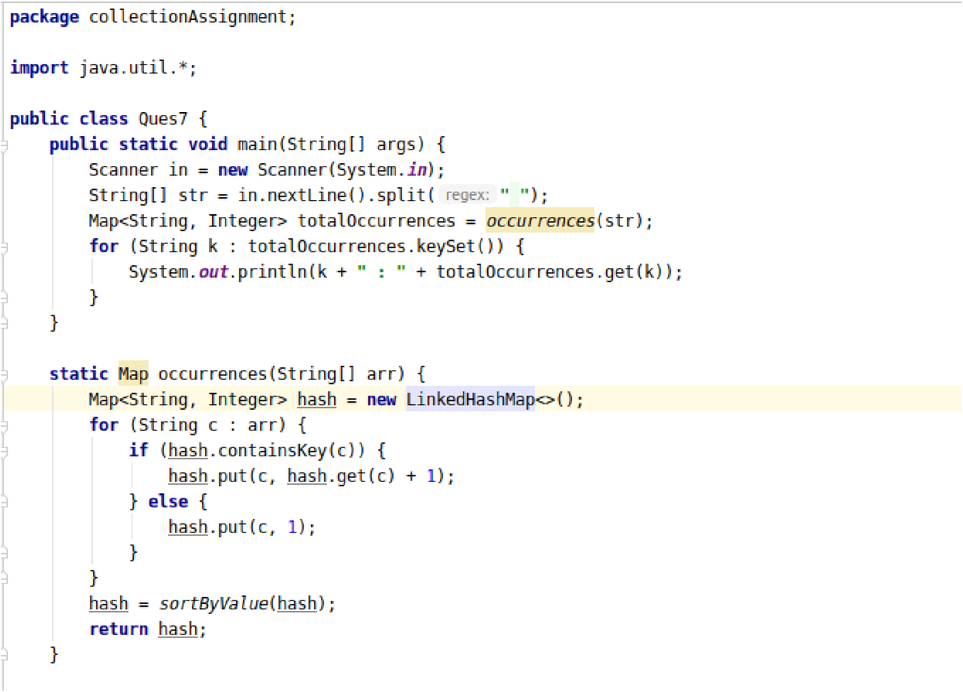
```java
        for (Student student : studList) {
            System.out.println(student);
        }

        Collections.sort(studList, new Comparator<Student>() {
            @Override
            public int compare(Student o1, Student o2) {
                int ret = 0;
                if(o1.score==o2.score){
                    ret = o1.name.compareTo(o2.name);
                }
                ret = o1.score > o2.score ? 1 : -1;
                return ret;
            }
        });

        System.out.println("-------------------------------------------");
        System.out.println("Sorted");
        System.out.println("-------------------------------------------");

        for (Student student : studList) {
            System.out.println(student);
        }

    }
}
```



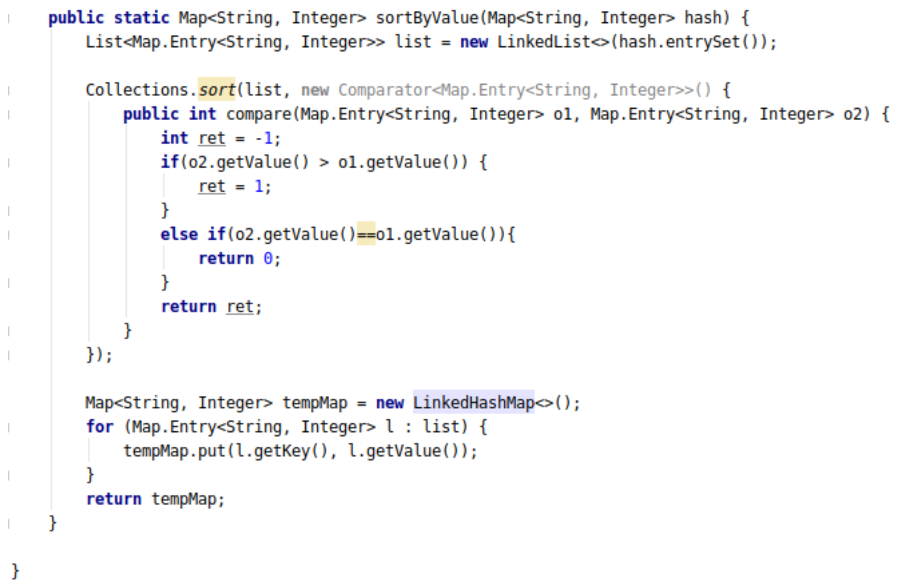
```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Student{name='Name1', score=31.0, age=32.0}
Student{name='Name2', score=35.0, age=38.0}
Student{name='Name3', score=38.0, age=45.0}
Student{name='Name4', score=33.0, age=25.0}
Student{name='Name5', score=36.0, age=46.0}
Student{name='Name6', score=24.0, age=34.0}
Student{name='Name7', score=32.0, age=48.0}
Student{name='Name8', score=22.0, age=46.0}
Student{name='Name9', score=34.0, age=46.0}
Student{name='Name10', score=32.0, age=25.0}
-------------------------------------------
Sorted
-------------------------------------------
Student{name='Name8', score=22.0, age=46.0}
Student{name='Name6', score=24.0, age=34.0}
Student{name='Name1', score=31.0, age=32.0}
Student{name='Name10', score=32.0, age=25.0}
Student{name='Name7', score=32.0, age=48.0}
Student{name='Name4', score=33.0, age=25.0}
Student{name='Name9', score=34.0, age=46.0}
Student{name='Name2', score=35.0, age=38.0}
Student{name='Name5', score=36.0, age=46.0}
Student{name='Name3', score=38.0, age=45.0}

Process finished with exit code 0
```

Question: 7:

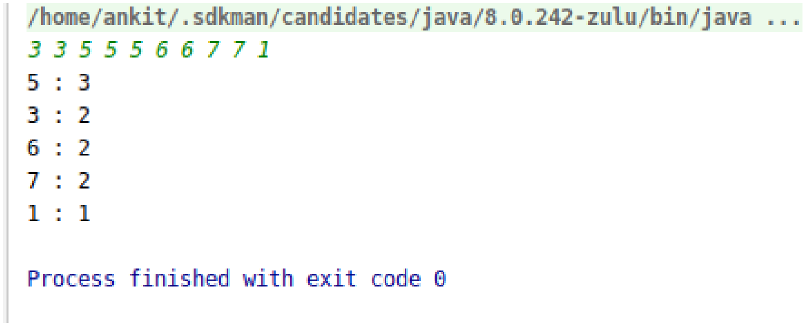
Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.



```java
package collectionAssignment;

import java.util.*;

public class Ques7 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String[] str = in.nextLine().split( regex: " ");
        Map<String, Integer> totalOccurrences = occurrences(str);
        for (String k : totalOccurrences.keySet()) {
            System.out.println(k + " : " + totalOccurrences.get(k));
        }
    }

    static Map occurrences(String[] arr) {
        Map<String, Integer> hash = new LinkedHashMap<>();
        for (String c : arr) {
            if (hash.containsKey(c)) {
                hash.put(c, hash.get(c) + 1);
            } else {
                hash.put(c, 1);
            }
        }
        hash = sortByValue(hash);
        return hash;
    }
```



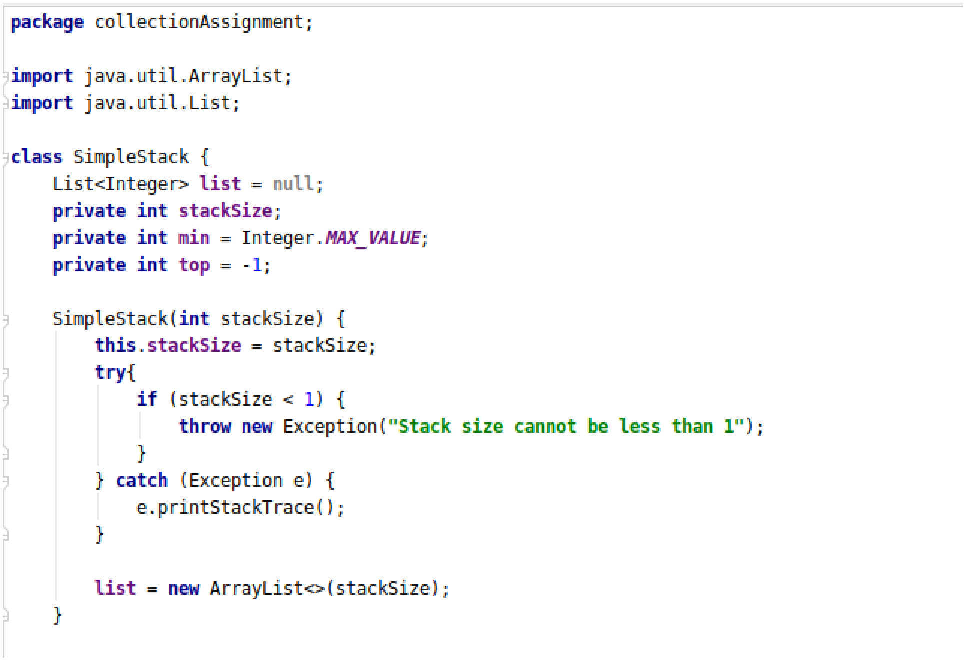
```
    public static Map<String, Integer> sortByValue(Map<String, Integer> hash) {
        List<Map.Entry<String, Integer>> list = new LinkedList<>(hash.entrySet());

        Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
                int ret = -1;
                if(o2.getValue() > o1.getValue()) {
                    ret = 1;
                }
                else if(o2.getValue()==o1.getValue()){
                    return 0;
                }
                return ret;
            }
        });

        Map<String, Integer> tempMap = new LinkedHashMap<>();
        for (Map.Entry<String, Integer> l : list) {
            tempMap.put(l.getKey(), l.getValue());
        }
        return tempMap;
    }

}
```



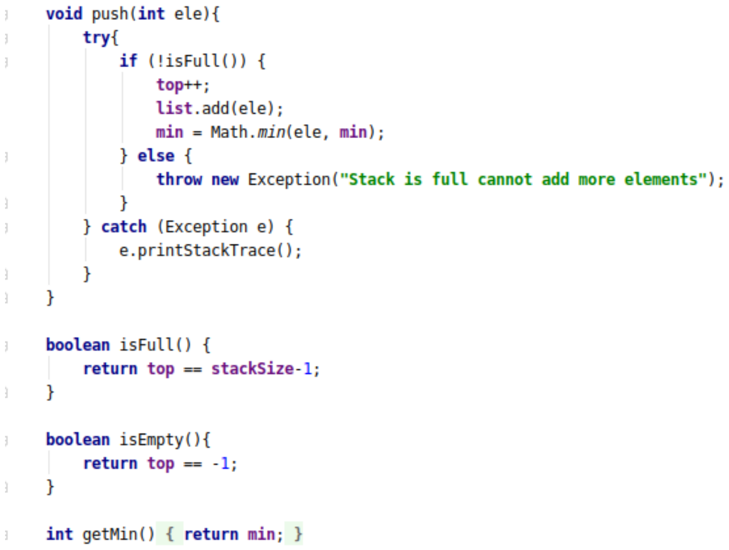
```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
3 3 5 5 5 6 6 7 7 1
5 : 3
3 : 2
6 : 2
7 : 2
1 : 1

Process finished with exit code 0
```

Question: 8:

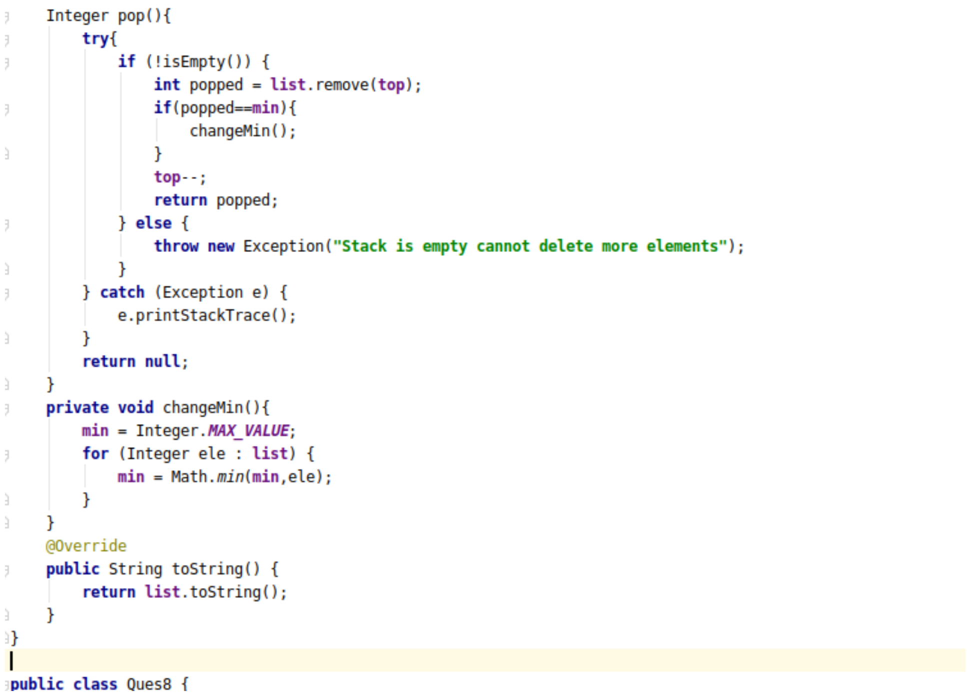
Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity O(1))



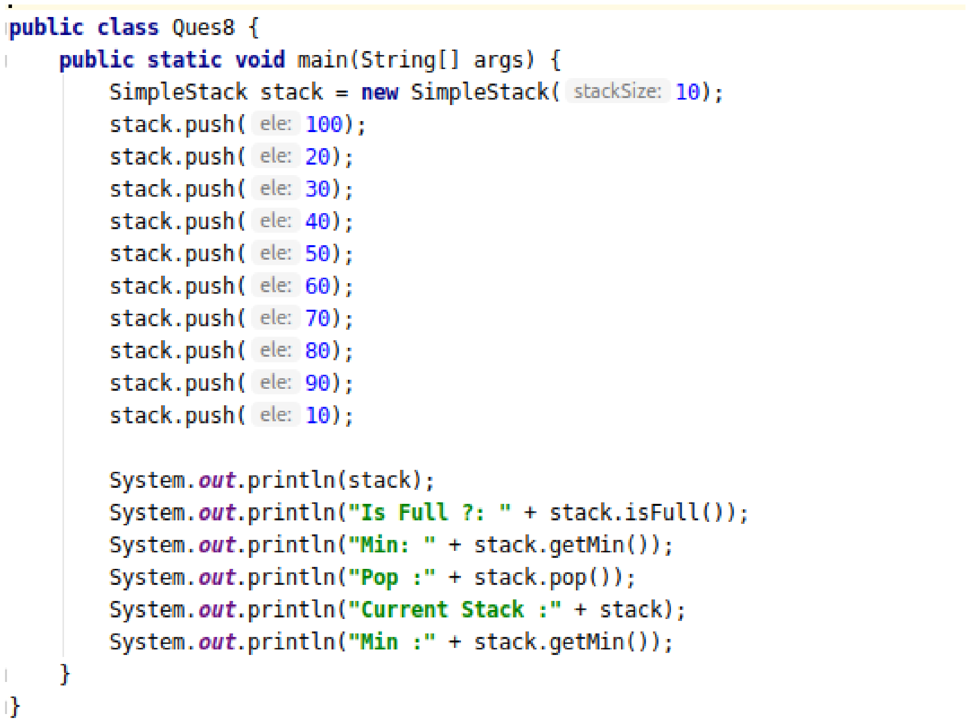
```java
package collectionAssignment;

import java.util.ArrayList;
import java.util.List;

class SimpleStack {
    List<Integer> list = null;
    private int stackSize;
    private int min = Integer.MAX_VALUE;
    private int top = -1;

    SimpleStack(int stackSize) {
        this.stackSize = stackSize;
        try{
            if (stackSize < 1) {
                throw new Exception("Stack size cannot be less than 1");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        list = new ArrayList<>(stackSize);
    }
```



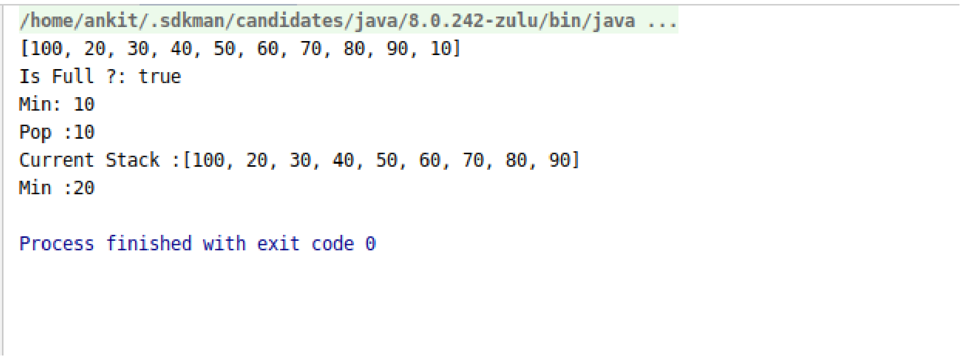
```java
    void push(int ele){
        try{
            if (!isFull()) {
                top++;
                list.add(ele);
                min = Math.min(ele, min);
            } else {
                throw new Exception("Stack is full cannot add more elements");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    boolean isFull() {
        return top == stackSize-1;
    }

    boolean isEmpty(){
        return top == -1;
    }

    int getMin() { return min; }
```



```java
    Integer pop(){
        try{
            if (!isEmpty()) {
                int popped = list.remove(top);
                if(popped==min){
                    changeMin();
                }
                top--;
                return popped;
            } else {
                throw new Exception("Stack is empty cannot delete more elements");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
    private void changeMin(){
        min = Integer.MAX_VALUE;
        for (Integer ele : list) {
            min = Math.min(min,ele);
        }
    }
    @Override
    public String toString() {
        return list.toString();
    }
}

public class Ques8 {
```



```java
public class Ques8 {
    public static void main(String[] args) {
        SimpleStack stack = new SimpleStack( stackSize: 10);
        stack.push( ele: 100);
        stack.push( ele: 20);
        stack.push( ele: 30);
        stack.push( ele: 40);
        stack.push( ele: 50);
        stack.push( ele: 60);
        stack.push( ele: 70);
        stack.push( ele: 80);
        stack.push( ele: 90);
        stack.push( ele: 10);

        System.out.println(stack);
        System.out.println("Is Full ?: " + stack.isFull());
        System.out.println("Min: " + stack.getMin());
        System.out.println("Pop :" + stack.pop());
        System.out.println("Current Stack :" + stack);
        System.out.println("Min :" + stack.getMin());
    }
}
```



```
/home/ankit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
[100, 20, 30, 40, 50, 60, 70, 80, 90, 10]
Is Full ?: true
Min: 10
Pop :10
Current Stack :[100, 20, 30, 40, 50, 60, 70, 80, 90]
Min :20

Process finished with exit code 0
```