

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.

### Sol 1.

```
1 package JavaDay4.Ques1;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5
6 public class App {
7     public static void main(String[] args) {
8         ArrayList<Float> sumList=new ArrayList<>();
9         sumList.add(2.1f);
10        sumList.add(1f);
11        sumList.add(8.55f);
12        sumList.add(7.3f);
13        sumList.add(4f);
14        float sum=0;
15        Iterator<Float> it= sumList.iterator();
16
17        while(it.hasNext()){
18            sum+=it.next();
19        }
20
21        System.out.println("The sum of list is: "+sum);
22    }
23 }
24
```

```
App x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
The sum of list is: 22.95

Process finished with exit code 0

Run Debug TODO
```

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

### Sol 2.

```
2 import java.util.*;
3 public class App {
4
5     public static void main(String[] args) {
6
7         boolean flag = false;
8         String test = "Trying string in TTN";
9         System.out.println(test);
10        test=test.toLowerCase();
11        List<Character> uniqueList = new LinkedList<>();
12        List<Character> repeatedList = new LinkedList<>();
13        HashSet<Character> unique = new HashSet<>();
14        for (int i=0; i<test.length();i++){
15            flag = unique.add(test.charAt(i));
16            if (flag==true) {
17                uniqueList.add(test.charAt(i));
18            }
19            else
20                repeatedList.add(test.charAt(i));
21        }
22        for (int i=0;i<uniqueList.size();i++){
23            for (int j=0;j<repeatedList.size();j++){
24                if (uniqueList.get(i).equals(repeatedList.get(j))){
25                    uniqueList.remove(i);
26                } }
27        }
28        System.out.print("Unique characters are:");
29        System.out.println(uniqueList.size());
30        Iterator<Character> uniqueChars= uniqueList.iterator();
31        while (uniqueChars.hasNext()){
32            System.out.print(uniqueChars.next()+" ", " ");
33        }
34    }
35 }
```

```
App (2) x
/home/aman/.sdkman/candidates/java/8.0.242-jdk8/bin/java -IdeaProjects/core-java/src/JavaDay4/Ques2/App.java
Trying string in TTN
Unique characters are:2
y, s,
Process finished with exit code 0

inal 4: Run 5: Debug 6: TODO
```

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

### Sol 3.

```
1 package JavaDay4.Ques3;
2
3 import java.util.*;
4
5 public class App {
6     public static void main(String[] args) {
7
8         String str="This is TTN company";
9         System.out.println("String entered: "+str);
10        str=str.toLowerCase();
11        System.out.println("String in lowercase: "+str+"\n");
12        Map<Character,Integer> map=new LinkedHashMap<>();
13        for(char i:str.toCharArray()){
14            if(map.keySet().contains(i))
15            {
16                int x=map.get(i);
17                map.put(i,++x);
18            }
19            else {
20                map.put(i,1); }
21        }
22        for (char c:map.keySet())
23        {
24            if(c==' ')
25                System.out.print(" ");
26            else
27                System.out.println(c+" "+map.get(c));
28        }
29    }
30 }
31 }
```

```
App (4) <
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
String entered: This is TTN company
String in lowercase: this is ttn company

t 3
h 1
i 2
s 2
n 2
c 1
o 1
m 1
p 1
a 1
y 1

Process finished with exit code 0

Run Debug TODO
```

4. Write a program to sort HashMap by value.

#### Sol 4.

```
1 package JavaDay4.Ques4;
2 import java.util.*;
3
4 public class App {
5     public static
6     Map<String, Integer> map = new HashMap<String, Integer>();
7
8     public static void main(String[] args) {
9
10         map.put("Cat", 20);
11         map.put("Dog", 100);
12         map.put("Swan", 40);
13         map.put("Giraffe", 1);
14         map.put("Tiger", 41);
15         map.put("Monkey", 26);
16
17         printMap(map);
18         System.out.println("\n");
19         //sortMap(map);
20
21         printMap(sortMap(map));
22
23     }
24 }
```

```
25
26 @ private static HashMap<String, Integer> sortMap(Map<String, Integer> map) {
27
28     List<Map.Entry<String, Integer>> list = new LinkedList<Map.Entry<String, Integer>>(map.entrySet());
29     Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
30
31         @Override
32         public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
33             return o1.getValue().compareTo(o2.getValue());
34         }
35     });
36
37     HashMap<String, Integer> sortedMap = new LinkedHashMap<String, Integer>();
38     for (Map.Entry<String, Integer> map1 : list) {
39         sortedMap.put(map1.getKey(), map1.getValue());
40     }
41
42     return sortedMap;
43 }
44
45 @ private static void printMap(Map map) {
46     Iterator<String> keyIterator = map.keySet().iterator();
47     Iterator<Integer> valueIterator = map.values().iterator();
48     while (keyIterator.hasNext() && valueIterator.hasNext()) {
49         System.out.println(keyIterator.next() + ": " + valueIterator.next());
50     }
51 }
52 }
```

```
App (5) x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/ ~/IdeaProjects/core-java/src/JavaDay4/Ques4/App.java
Swan: 40
Cat: 20
Monkey: 26
Dog: 100
Tiger: 41
Giraffe: 1

Giraffe: 1
Cat: 20
Monkey: 26
Swan: 40
Tiger: 41
Dog: 100

Process finished with exit code 0
```

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

### Sol 5.

```
1 package JavaDay4.Ques5;
2
3 import java.util.*;
4 import java.util.ArrayList;
5 import java.util.Collections;
6 import java.util.Comparator;
7
8 class Test{
9
10     String name;
11     int salary;
12     int age;
13
14     public Test(String name, int salary, int age) {
15         this.name = name;
16         this.salary = salary;
17         this.age = age;
18     }
19
20     @Override
21     public String toString() {
22         return "Employee{" +
23             "name='" + name + '\'' +
24             ", salary=" + salary +
25             ", age=" + age +
26             '}';
27     }
28 }
29
30 class SortBySalary implements Comparator<Test>{
31
32     Employee main()
```

```
29
30 class SortBySalary implements Comparator<Test>{
31
32     @Override
33     public int compare(Test o1, Test o2) {
34         return o1.salary - o2.salary;
35     }
36 }
37
38 public class Employee{
39     public static void main(String[] args) {
40         ArrayList<Test> list = new ArrayList<Test>();
41         list.add(new Test( "A", salary: 20000, age: 22));
42         list.add(new Test( "B", salary: 10000, age: 21));
43         list.add(new Test( "S", salary: 10000, age: 22));
44         list.add(new Test( "T", salary: 15000, age: 21));
45
46         System.out.println("Unsorted");
47         for (int i=0; i<list.size(); i++)
48             System.out.println(list.get(i));
49
50         Collections.sort(list, new SortBySalary());
51
52         System.out.println("\nSorted by salary");
53         for (int i=0; i<list.size(); i++)
54             System.out.println(list.get(i));
55     }
56 }
57
58
59 Employee main()
```

```
Employee x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Unsorted
Employee(name='A', salary=20000, age=22)
Employee(name='B', salary=10000, age=21)
Employee(name='S', salary=10000, age=22)
Employee(name='T', salary=15000, age=21)

Sorted by salary
Employee(name='B', salary=10000, age=21)
Employee(name='S', salary=10000, age=22)
Employee(name='T', salary=15000, age=21)
Employee(name='A', salary=20000, age=22)

Process finished with exit code 0
```

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; Int Score; Int Age}

### Sol 6.

```
1 package JavaDay4.Ques6;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.Comparator;
6
7 public class Student{
8     String Name;
9     int Score;
10    int age;
11
12    public Student(String name, int score, int age) {
13        Name = name;
14        Score = score;
15        this.age = age;
16    }
17
18
19
20    @Override
21    public String toString() {
22        return "Student{" +
23            "Name='" + Name + '\'' +
24            ", Score=" + Score +
25            ", age=" + age +
26            '}';
27    }
28 }
```

```
29
30 class Sort implements Comparator<Student>{
31
32    @Override
33    public int compare(Student o1, Student o2) {
34        if (o1.Score == o2.Score){
35            return o1.Name.compareTo(o2.Name);
36        }else
37            return o1.Score - o2.Score;
38    }
39 }
40
```

```
41 class Main {
42
43     public static ArrayList<Student> list = new ArrayList<Student>();
44
45     public static void main(String[] args) {
46
47         list.add(new Student( name: "Aman", score: 99, age: 21));
48         list.add(new Student( name: "Anirudh", score: 99, age: 21));
49         list.add(new Student( name: "Sudha", score: 99, age: 21));
50         list.add(new Student( name: "Sahil", score: 92, age: 21));
51         list.add(new Student( name: "Jeet", score: 90, age: 21));
52
53
54         System.out.println("Student List: \n");
55         for (int i=0; i<list.size(); i++){
56             System.out.println(list.get(i));
57         }
58
59         System.out.println("\n");
60         Collections.sort(list, new Sort());
61
62         System.out.println("Sorted list based on students score:"+'\n');
63         for (int i=0; i<list.size();i++){
64             System.out.println(list.get(i));
65         }
66     }
67 }
68
69 }
```

```
Main
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Student List:

Student{Name='Aman', Score=99, age=21}
Student{Name='Anirudh', Score=99, age=21}
Student{Name='Sudha', Score=99, age=21}
Student{Name='Sahil', Score=92, age=21}
Student{Name='Jeet', Score=90, age=21}

Sorted list based on students score:

Student{Name='Jeet', Score=90, age=21}
Student{Name='Sahil', Score=92, age=21}
Student{Name='Aman', Score=99, age=21}
Student{Name='Anirudh', Score=99, age=21}
Student{Name='Sudha', Score=99, age=21}

Process finished with exit code 0
```

inal Messages Find Run Debug TODO

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

### Sol 7.

```
1 package JavaDay4.Ques7;
2 import java.util.*;
3 public class App {
4     @private static void sortedArray(int[] inputArray)
5     {
6         Map<Integer, Integer> map = new LinkedHashMap<>();
7         for (int i = 0; i < inputArray.length; i++)
8         {
9             if (map.containsKey(inputArray[i]))
10             {
11                 map.put(inputArray[i], map.get(inputArray[i])+1); }
12             else
13             {
14                 map.put(inputArray[i], 1); }
15         }
16         ArrayList<Integer> sortedElements = new ArrayList<>();
17         map.entrySet().stream().sorted(Collections.reverseOrder(Map.Entry.comparingByValue()))
18             .forEach(entry -> {
19                 for(int i = 1; i <= entry.getValue(); i++)
20                     sortedElements.add(entry.getKey());
21             });
22         System.out.println("Unsorted Array Elements: ");
23         System.out.println("Input Array :"+ Arrays.toString(inputArray) +"\n");
24         System.out.println("Sorted Array Elements In Descending Order Of their Frequency: ");
25         System.out.println(sortedElements);
26     }
27     public static void main(String[] args)
28     { sortedArray(new int[] {7, 1, 3, 4, 7, 1, 7, 1, 4, 5, 1, 9, 3}); }
29 }
30
31
```

```
App x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Unsorted Array Elements:
Input Array :[7, 1, 3, 4, 7, 1, 7, 1, 4, 5, 1, 9, 3]

Sorted Array Elements In Descending Order Of their Frequency:
[1, 1, 1, 1, 7, 7, 7, 3, 3, 4, 4, 5, 9]

Process finished with exit code 0

inal  Q 3: Find  ▶ 4: Run  ⚙ 5: Debug  ☰ 6: TODO
```



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  $O(1)$ )

### Sol 8.

```
1 package JavaDay4.Ques8;
2
3 import java.util.Stack;
4
5 public class App extends SpecialStack {
6     public static void main(String[] args) {
7         SpecialStack specialStack = new SpecialStack();
8
9         specialStack.push( element: 9);
10        specialStack.push( element: 5);
11        specialStack.push( element: 10);
12        specialStack.pop();
13        specialStack.getMin();
14        specialStack.push( element: 1);
15        specialStack.push( element: 6);
16        specialStack.getMin();
17        specialStack.pop();
18    }
19 }
20
```

```
1 class SpecialStack{
2     Stack<Integer> stack;
3     Integer minElement;
4
5     public SpecialStack() {
6         stack = new Stack<Integer>();
7     }
8
9     void getMin(){
10        if(stack.isEmpty()){
11            System.out.println("Stack is Empty.");
12        }else {
13            System.out.println("Minimum Element in the stack is: "+minElement);
14        }
15    }
16
17    void peek(){
18        if(stack.isEmpty()){
19            System.out.println("Stack is Empty.");
20            return;
21        }
22
23        Integer top = stack.peek();
24        System.out.println("Top most element of the stack is: ");
25
26        if (top<minElement){
27            System.out.println(minElement);
28        }else {
29            System.out.println(top);
30        }
31    }
32 }
```

```

53     void pop(){
54         if(stack.isEmpty()){
55             System.out.println("Stack is Empty.");
56             return;
57         }
58
59         System.out.println("Top element removed: ");
60         Integer top = stack.pop();
61
62         if (top==minElement){
63             System.out.println(minElement);
64             minElement = 2*minElement-top;
65         }
66         else
67             System.out.println(top);
68     }
69     void push(Integer element){
70         if (stack.isEmpty())
71         {
72             minElement = element;
73             stack.push(element);
74             System.out.println("Number Inserted: " + element);
75             return;
76         }
77
78         if (element < minElement)
79         {
80             stack.push( item: 2*element - minElement);
81             minElement = element;
82         }

```

```

83     else
84         stack.push(element);
85
86     System.out.println("Number Inserted: " + element);
87 }
88
89
90 }

```

SpecialStack > push()

```

App (1) x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Number Inserted: 9
Number Inserted: 5
Number Inserted: 10
Top element removed:
10
Minimum Element in the stack is: 5
Number Inserted: 1
Number Inserted: 6
Minimum Element in the stack is: 1
Top element removed:
6
Process finished with exit code 0

```

nal Messages Find Run Debug TODO