IRC_SKCT_Java2_COD_Stream

Test Summary

- No. of Sections: 1
- No. of Questions: 10
- Total Duration: 120 min

Section 1 - Coding

Section Summary

- No. of Questions: 10
- Duration: 120 min

Additional Instructions:

None

Q1. Helen is having a list of integers, your task is to help her find the maximum value element present in the list using Stream functions

Input Format

First line of input has the number of Integers(N) that will be given to Helen
Second Line has the first integer value
Third Line has the second Integer value and so on upto N

Output Format

Maximum number from all the integers

Sample Input

```
3
546
546
12
```

Sample Output

```
546
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. Given a list of integers, find the total number of elements present in the list using Stream functions

Input Format

First Line has the number of integer(N)
Following lines has N values

Output Format

Total number of elements that is stored in the List

Sample Input

```
3
123
466
899
```

Sample Output

```
3
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. Vikram is to find duplicate elements in each integers list in java using Stream functions. Help him pick only the duplicate elements from the list.

Input Format

First Line has the Integer N(number of elements)
Followed by the N integer values

Output Format

First Integer Value of the given elements

Sample Input

Sample Output

3 45 41 45	45

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. Gokul is given a list of N integers, help him find out all the even numbers exist in the list using Stream functions

Input Format

First Line has the number of Integers N
Followed by N values

Output Format

Even Integer numbers

Sample Input **Sample Output**

3 2224 2225 2222	2224 2222

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. Athiyan is having a list of integers, your task is to help him find out all the numbers starting with 1 using Stream functions

Input Format

First Line has a Integer N
Followed by N values to be processed

Output Format

Integer Values starting with 1

Sample Input **Sample Output**

4 12 23 14	12 14 12

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6. Koutheesh is having a list of integers, help him to sort all the values present in it using Stream functions

Input Format

First line has the number of Integer values **N** to be sorted
Followed by N values

Output Format

Sorted Integer values from the List

Sample Input **Sample Output**

5 45 54 65	12 45 45 54

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q7. Sakthivel is supposed to filter out all the words that exceed n letters, then sum up the lengths of all such words exceeding n letters

Input Format

First line of input has the number of words given(M)
Second line has all the M words
Third Line has the value of n

Output Format

Sum of length of words exceeding n

Sample Input

```
3
Gokul Sam Prasath
4
```

Sample Output

```
12
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q8. Einstein is trying to count the number of specific colored balls from a bucket, all the balls are colored from a to z. He is supposed to find the number of **C** colored balls(i.e., One color from a-z) using Stream API methods.

Input Format

First Line consists of String **S**(Bucket of Balls)
Second Line has a alphabet Character **C**(one character from a-z)

Output Format

Integer or Long value mentioning the count of **C in S**

Sample Input

```
Gokul is in God mode
o
```

Sample Output

```
3
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q9. **Problem statement:**
Create a class named " Product" to have the details of id, name, and price of the product details. Write a program to compute the sum of all the product prices using the collectors class and its specified methods.

Input Format

The input consists of two integers, one string, and one float.
First Integer gets the number of products.
The second Integer gets the id of the product.
Third Integer to get the name of the product.
The fourth one is to get the price of the product.

Output Format

The output consists of the sum of all the product prices.

Sample Input

```
2
1
hp
28000
```

Sample Output

```
58000.0
```

Sample Input

```
3
1
Azus
20000
```

Sample Output

```
61000.0
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q10. **Problem statement:**
Write a program for the Fibonacci series using streams

Input Format

The input consists of one Integer

Output Format

The output consists of series of numbers,
Refer the sample input and output for specifications.

Sample Input

Sample Output

5	[0, 1, 1, 2, 3]

Sample Input **Sample Output**

6	[0, 1, 1, 2, 3, 5]

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
5
1
2
3
```

Output

```
5
```

Weightage - 25

Input

```
6
78
98
45
```

Output

```
98
```

Weightage - 25

Input

```
3
45687
54689
12345
```

Output

```
54689
```

Weightage - 25

Input

```
2
900000
900001
```

Output

```
900001
```

Weightage - 25

Sample Input

```
3
546
546
12
```

Sample Output

```
546
```

Solution

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

class main {

    public static void main(String[] args) {
        List<Integer> arList = new ArrayList<Integer>();
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        for(int i=0;i<num;i++)
        {
            arList.add(sc.nextInt());
        }
```

```
        int max =  arList.stream()
                        .max(Integer::compare)
                        .get();
        System.out.println(max);
    }

}
```

Q2

Test Case

Input

```
5
123
456
789
```

Output

```
5
```

Weightage - 25

Input

```
7
1
2
3
```

Output

```
7
```

Weightage - 25

Input

```
4
654
654
654
```

Output

```
4
```

Weightage - 25

Input

```
9
12
21
23
```

Output

```
9
```

Weightage - 25

Sample Input

```
3
123
466
899
```

Sample Output

```
3
```

Solution

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

class main {
    public static void main(String[] args) {
        List<Integer> arList = new ArrayList<Integer>();
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
```

```
        for(int i=0;i<num;i++)
        {
            arList.add(sc.nextInt());
        }
        long count =  arList.stream()
                      .count();
        System.out.println(count);
    }
}
```

Q3

Test Case

Input

```
5
45
54
45
```

Output

```
45
```

Weightage - 25

Input

```
6
54
65
54
```

Output

```
54
65
23
```

Weightage - 25

Input

```
2
895
895
```

Output

```
895
```

Weightage - 25

Input

```
8
45
65
12
```

Output

```
65
12
```

Weightage - 25

Sample Input

```
3
45
41
45
```

Sample Output

```
45
```

Solution

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Scanner;
import java.util.Set;

class main{
```

```
    public static void main(String args[]) {
        List<Integer> arList = new ArrayList<Integer>();
        Set<Integer> set = new HashSet<Integer>();
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        for(int i=0;i<num;i++)
        {
            arList.add(sc.nextInt());
        }
        arList.stream()
              .filter(n -> !set.add(n))
              .forEach(System.out::println);
    }
}
```

Q4

Test Case

Input

```
6
24
51
24
```

Output

```
24
24
```

Weightage - 25

Input

```
4
79
12
2
```

Output

```
12
2
```

Weightage - 25

Input

```
3
45
54
56
```

Output

```
54
56
```

Weightage - 25

Input

```
10
45
65
78
```

Output

```
78
546
154
222
```

Weightage - 25

Sample Input

```
3
2224
2225
2222
```

Sample Output

```
2224
2222
```

Solution

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

```
import java.util.List;
class main {
            public static void main(String args[]) {
                    List<Integer> arList = new ArrayList<Integer>();
                    Scanner sc=new Scanner(System.in);
                    int num=sc.nextInt();
                    for(int i=0;i<num;i++)
                    {
                        arList.add(sc.nextInt());
                    }
                    arList.stream()
                          .filter(n -> n%2 == 0)
                          .forEach(System.out::println);
            }
}
```

Q5

Test Case

Input	Output
5 54 65 12	12

Weightage - 25

Input	Output
3 12 12 12	12 12 12

Weightage - 25

Input	Output
6 12 13 45	12 13

Weightage - 25

Input	Output
2 1 1	1 1

Weightage - 25

Sample Input	Sample Output
4 12 23 14	12 14 12

Solution

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

class main {
    public static void main(String[] args) {
        List<Integer> arList = new ArrayList<Integer>();
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        for(int i=0;i<num;i++)
        {
            arList.add(sc.nextInt());
        }
        arList.stream()
              .map(s -> s + "") // Convert integer to String
              .filter(s -> s.startsWith("1"))
              .forEach(System.out::println);
    }
}
```

Q6

Test Case

Input

```
4
5
4
3
```

Output

```
2
3
4
5
```

Weightage - 25

Input

```
6
45
46
47
```

Output

```
1
45
46
47
```

Weightage - 25

Input

```
3
5
2
4
```

Output

```
2
4
5
```

Weightage - 25

Input

```
8
45
65
12
```

Output

```
12
45
65
123
```

Weightage - 25

Sample Input

```
5
45
54
```

Sample Output

```
12
45
45
```

Solution

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

class main {
    public static void main(String[] args) {
    List<Integer> arList = new ArrayList<Integer>();
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        for(int i=0;i<num;i++)
        {
            arList.add(sc.nextInt());
        }
        arList.stream()
              .sorted()
              .forEach(System.out::println);
}
```

Q7

Test Case

Input

```
3
Kouthish Sus Sakthi
3
```

Output

```
14
```

Weightage - 25

Input

```
5
world is not enough 007
4
```

Output

```
11
```

Weightage - 25

Input

```
2
Vikram Karnan
5
```

Output

```
12
```

Weightage - 25

Input

```
4
hi! how are you
3
```

Output

```
0
```

Weightage - 25

Sample Input

Sample Output

3 Gokul Sam Prasath 4	12

Solution

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

class main {

    public static void main(String[] args) {
        List<String> arList = new ArrayList<String>();
        Scanner sc=new Scanner(System.in);
        int num=sc.nextInt();
        for(int i=0;i<num;i++)
        {
            arList.add(sc.next());
        }
        int n=sc.nextInt();
        int ans=arList.stream()
                    .filter(s -> s.length() > n)
                    .mapToInt(s -> s.length())
                    .sum();
        System.out.println(ans);
    }
}
```

Q8

Test Case

Input **Output**

Iamneo is a fair place to work with w	2

Weightage - 25

Input **Output**

Java is becoming an Omnivore a	3

Weightage - 25

Input **Output**

Generate input via Code n	2

Weightage - 25

Input **Output**

Maximum Allowed Memory limit l	3

Weightage - 25

Sample Input

```
Gokul is in God mode
o
```

Sample Output

```
3
```

Solution

```
import java.util.Scanner;

class main {
        public static void main(String args[])
        {
            Scanner sc=new Scanner(System.in);
            String str = sc.nextLine();
            char c =sc.next().charAt(0);
            Long l= str.chars()
                    .filter(val -> val == c)
                    .count();
            System.out.println(l);
        }
}
```

Q9

Test Case

Input

```
2
1
Lenovo
28000
```

Output

```
48000.0
```

Weightage - 20

Input

```
3
1
Apple
30000
```

Output

```
63000.0
```

Weightage - 20

Input

```
2
1
Hcl
40000
```

Output

```
72000.0
```

Weightage - 20

Input

```
3
1
smartron
50000
```

Output

```
123000.0
```

Weightage - 20

Input

```
2
1
iBall
28000
```

Output

```
58000.0
```

Weightage - 20

Sample Input

```
2
1
hp
28000
```

Sample Output

```
58000.0
```

Sample Input

```
3
1
Azus
20000
```

Sample Output

```
61000.0
```

Solution

```
import java.util.*;
import java.util.stream.Collectors;
class Product
{
    int id;
    String name;
    float price;
    public Product(int id, String name, float price)
    {
        this.id = id;
        this.name = name;
        this.price = price;
    }
}
class Main
{
    public static void main(String[] args)
    {
        int n;
        Scanner  s = new Scanner(System.in);
        n=s.nextInt();
        List<Product> productsList = new ArrayList<Product>();
        for(int i = 0; i< n; i++)
        {
            int id =s.nextInt();
            String name=s.next();
            float price=s.nextFloat();
            productsList.add(new Product(id,name,price));
        }
        double totalPrice3 = productsList.stream()
                        .collect(Collectors.summingDouble(product->product.price));
        System.out.println(totalPrice3);

    }
}
```

Q10

Test Case

Input

```
10
```

Output

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Weightage - 20

Input

```
9
```

Output

```
[0, 1, 1, 2, 3, 5, 8, 13, 21]
```

Weightage - 20

Input

```
7
```

Output

```
[0, 1, 1, 2, 3, 5, 8]
```

Weightage - 20

Input

```
3
```

Output

```
[0, 1, 1]
```

Weightage - 20

Input

```
15
```

Output

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 2
```

Weightage - 20

Sample Input

```
5
```

Sample Output

```
[0, 1, 1, 2, 3]
```

Sample Input

```
6
```

Sample Output

```
[0, 1, 1, 2, 3, 5]
```

Solution

```
import java.util.*;
import java.util.List;
```

```
import java.util.stream.Stream;
import static java.util.stream.Collectors.toList;

class Main
{

    public static List<Integer> generate(int series) {
        return Stream.iterate(new int[]{0, 1}, s -> new int[]{s[1], s[0] + s[1]})
                .limit(series)
                .map(n -> n[0])
                .collect(toList());
    }

    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        int  num=s.nextInt();

        System.out.println(Main.generate(num));
    }
}
```