

Contents

Java – First Problem Statement	2
- Question.....	2
- Solution	2
- Test cases	3
Java – Second Problem Statement.....	4
- Question.....	4
- Solution	6
- Test cases	9

Solutions for TCS Xplore IPA held on 22-Apr-'23

Java – First Problem Statement

Question:

Write main method in Solution class.

In the main method, read a string and find the count of words starting with a vowel in the string. If no words are present in the String value then it should print "No String found".

Note:

All search should be case insensitive.

Sample input1:

Everyone should practice and learn to became professional.

Output:

2

Sample input2:

hi guys

Output:

No String found

Solution:

```
import java.util.Scanner;
public class Solution{
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        String s1=sc.nextLine();
        int vcount = 0;
        String[] s2 = s1.split(" ");
        for(int i=0;i<s2.length;i++)
        {
            char f = s2[i].charAt(0);
            if(f == 'A' || f=='E' || f=='I' || f=='O' || f=='U' || f=='a' || f=='e' || f=='i' || f=='o' || f=='u')
                vcount++;
        }
    }
}
```

2 of 11

2

Restricted for circulation outside TCS Xplore

```
    if(vcount>0)
        System.out.println(vcount);
    else
        System.out.println("No String found");
}
}
```

Test Cases

Test Case1:

Input:

Everyone should respect their elders.

Output:

2

Test Case2:

Input:

hi guys welcome

Output:

No String found

Test Case3:

Input:

If you are alone then feel the nature.

Output:

3

Test Case4:

Input:

The greatest glory in living lies not in never falling, but in rising every time we fall.

Output:

4

3 of 11

3

Restricted for circulation outside TCS Xplore

Question:

Create a class Resort with the below attributes:

resortId- int
resortName - String
category - String
price- double
rating - double

The above attributes should be private, write getters, setters and parameterized constructor as required.

Create class Solution with main method.

Implement a static method - findAvgPriceByCategory in Solution class.

findAvgPriceByCategory method:

This method will take two input parameters - array of Resort objects and String parameter.

The method will return the average price of Resort(as int value) from array of Resort objects for the given category(String parameter passed) and whose rating is greater than 4.

If no Resort with the above condition is present in the array of Resort objects, then the method should return 0.

Note : All the searches should be case insensitive.

The above mentioned static method should be called from the main method.

For findAvgPriceByCategory method - The main method should print the returned average price of Resort as it is, if the returned value is greater than 0 otherwise it should print "There are no such available resort"

Eg: Average price of the 3 Star Resort: 9250
where 9250 is the average price and 3 Star is the category.

4 of 11

4

Restricted for circulation outside TCS Xplore

Before calling these static methods in main, use Scanner object to read the values of four Resort objects referring attributes in the above mentioned attribute sequence.
Next, read one String value for capturing category input.

Consider below sample input and output:

Testcase1:

Input:

1005
Samudra
3 star
3500.00
3.5
1001
O by Tamara
5 Star
7500.00
4
1007
Edens resort
3 Star
2500.00
4.7
1003
Tea Valley
3 Star
4600.00
4.3
3 Star

Output:

Average price of the 3 Star Resort:3550

Testcase2:

Input:

1005
Samudra
3 star
3500.00
3.5
1001
- . -

```
7500.00
4
1007
Edens resort
3 Star
2500.00
4.7
1003
Tea Valley
3 Star
4600.00
4.3
2 Star
```

Output:

There are no such available resort

Solution:

```
import java.util.Scanner;
public class Solution{
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Resort[] resort=new Resort[4];
        Scanner sc = new Scanner(System.in);
        for (int i = 0; i < resort.length; i++) {

            int resortId = sc.nextInt();
            sc.nextLine();
            String name = sc.nextLine();
            String category = sc.nextLine();
            double price = sc.nextDouble();
            double rating=sc.nextDouble();
            sc.nextLine();
            resort[i] = new Resort(resortId, name, category, price, rating);
        }
        String inp_category=sc.nextLine();
        int averagePrice = findAvgPriceByCategory (resort, inp_category);
        if (averagePrice > 0) {
```

```
        System.out.println("Average price of the "+inp_category+
Resort:"+averagePrice);
    } else {
        System.out.println("There are no such available resort");
    }
}
public static int findAvgPriceByCategory(Resort[] resort, String category) {
    int avgAmount = 0;
    int count = 0;
    int avgPrice = 0;
    for (int i = 0; i < resort.length; i++) {
        if (resort[i].getCategory().equalsIgnoreCase(category) && resort[i].getRating()>4
{
            avgPrice += resort[i].getPrice();
            count++;
        }
    }
    if (count > 0) {
```

```

        avgAmount = avgPrice / count;
    }
    return avgAmount;
}

}

class Resort{
    private int resortId;
    private String name;
    private String category;
    private double price;
    private double rating;
    public int getResortId() {
        return resortId;
    }
    public void setResortId(int resortId) {
        this.resortId = resortId;
    }
    public String getName() {
        return name;
    }
}

```

7 of 11

7

Restricted for circulation outside TCS Xplore

```

    }
    public void setName(String name) {
        this.name = name;
    }
    public String getCategory() {
        return category;
    }
    public void setCategory(String category) {
        this.category = category;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public double getRating() {
        return rating;
    }
    public void setRating(double rating) {
        this.rating = rating;
    }
    public Resort(int resortId, String name, String category, double price, double rating) {
        super();
        this.resortId = resortId;
        this.name = name;
        this.category = category;
        this.price = price;
        this.rating = rating;
    }
}

```

8 of 11

8

Restricted for circulation outside TCS Xplore

Test Cases:

Test Case 1:

Input:

1005
Samudra
3 star
3500.00
3.5
1001
O by Tamara
5 Star
7500.00
4
1007
Edens resort
3 Star
2500.00
4.7
1003
Tea Valley
3 Star
4600.00
4.3
3 Star

Output:

Average price of the 3 Star Resort:3550

Test Case 2:

Input:

1005
Samudra
3 star
3500.00
3.5
1001
O by Tamara
5 Star
7500.00

9 of 11

9

Restricted for circulation outside TCS Xplore

4
1007
Edens resort
3 Star
2500.00
4.7
1003
Tea Valley
3 Star
4600.00
4.3
2 Star

Output:

There are no such available resort

Test Case 3:

Input:

1005
ElSym Gardens
5 star
9040.00
4.2
1001
Emerald Inn
5 Star
8500.00
4.4
1007
Westend resort

5 Star
7900.00
4.1
1003
Greenridge resort
3 Star
4600.00
4.3
5 Star

10 of 11

10

Restricted for circulation outside TCS Xplore

Output:
Average price of the 5 Star Resort:8480

Test Case 4:

Input:
1005
Elsym Gardens
5 star
9040.00
4.2
1001
Emerald Inn
5 Star
8500.00
4.4
1007
Westend resort
2 Star
3900.00
3.9
1003
Greenridge resort
2 STAR
4600.00
4.3
2 Star

Output:
Average price of the 2 Star Resort:4600

11 of 11

11

Restricted for circulation outside TCS Xplore