Adventure Ascent(---RETIRED---)

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Based on: Adventure Ascent(---RETIRED---)
Run: Yes **Evaluate**: Yes
Automatic grade: Yes

Adventure Ascent is a renowned Trekking agency in the whole city. They wanted to count and get the mountain names based on the mountain peak point. The manager intimates a software developer to help in their process. You, being the software developer, develop a Java program based on the requirement.

Component Specification: TrekkingDetailsMain Class (Class)

Type (Class)	Attributes	Methods
TrekkingDetailsMain	private Map<String, Integer> **detailsMap**	Getters and setters methods for the attribute are included in the code skeleton.

Note: *Here the detailsMap, holds the Key as mountainName and Value as mountainPeakPoint.*

Requirement 1: Filter the mountains based on the minimumPeak and maximumPeak.

Type (Class)	Methods	Responsibilities
TrekkingDetailsMain	public int **findCountOfMountainsBasedOnThePeakPoint** (int minimumPeak, int maximumPeak)	This method accepts two parameters, minimumPeak, and maximumPeak. It filters and counts the number of mountains in the range and returns the result. Else return -1.

		Condition: Both minimumPeak and maximumPeak are inclusive

Requirement 2: Filter the mountain names based on the peak point.

Type (Class)	**Methods**	**Responsibilities**
TrekkingDetailsMain	public List<String> **findMountainsBasedOnPeakPoint**(int peakPoint)	This method accepts a parameter, peakPoint. Filter the mountain names based on the given peakPoint, and return the list of mountain names. ***Condition: All mountains whose peak point is less than or equal to the specified peakPoint.***

You are provided with the main method as code template and it is excluded from evaluation.

Note:

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user, and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question description.
- Ensure to provide the names for the classes, attributes, and methods as specified in the question description.
- Adhere to the code template, if provided.

Sample Input/Output 1:

Enter number of details to be added

3

Enter the details (Mountain name: Peak point)

Denali:78

MountElbert:450

MountArvon:345

Enter the minimum and maximum peak points

345

450

The total count of mountains are 2

Enter the peak point to be searched

100

Mountains based on the condition are

Denali

Sample Input/Output 2:

Enter number of details to be added

2

Enter the details (Mountain name: Peak point)

MountFressill:720

RockCandy:267

Enter the minimum and maximum peak points

200

250

No mountains were found

Enter the peak point to be searched

270

Mountains based on the condition are

RockCandy

Sample Input/Output 3:

Enter number of details to be added

2

Enter the details (Mountain name: Peak point)

MountFressill:720

RockCandy:267

Enter the minimum and maximum peak points

250

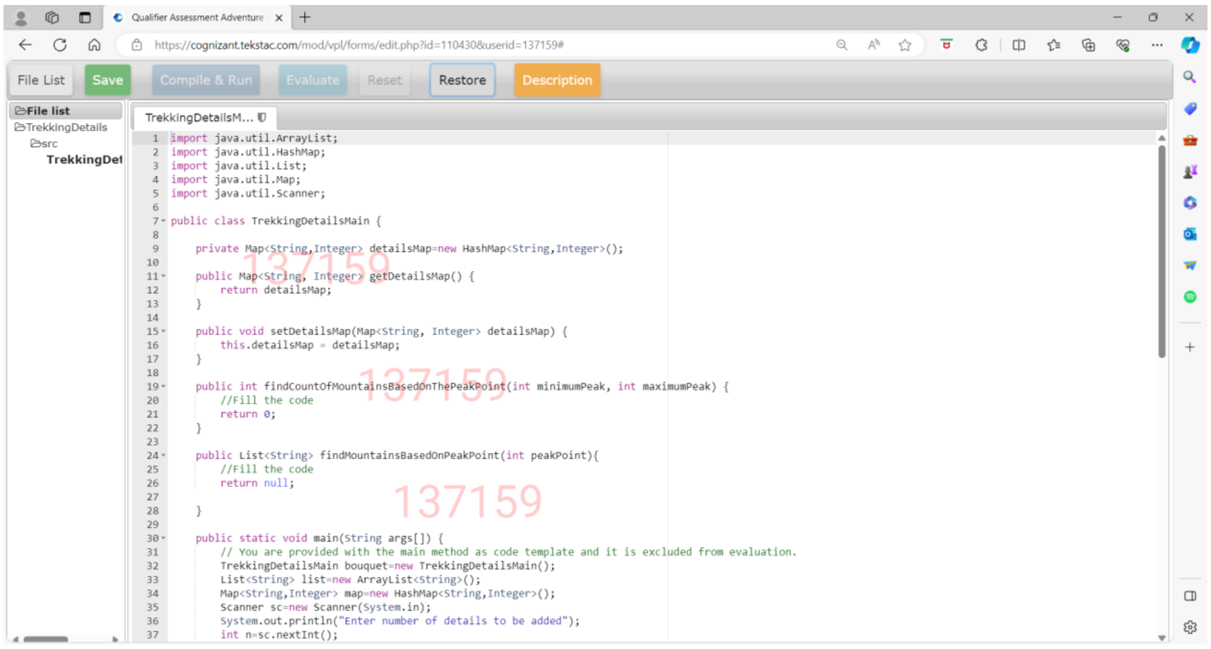
300

The total count of mountains are 1

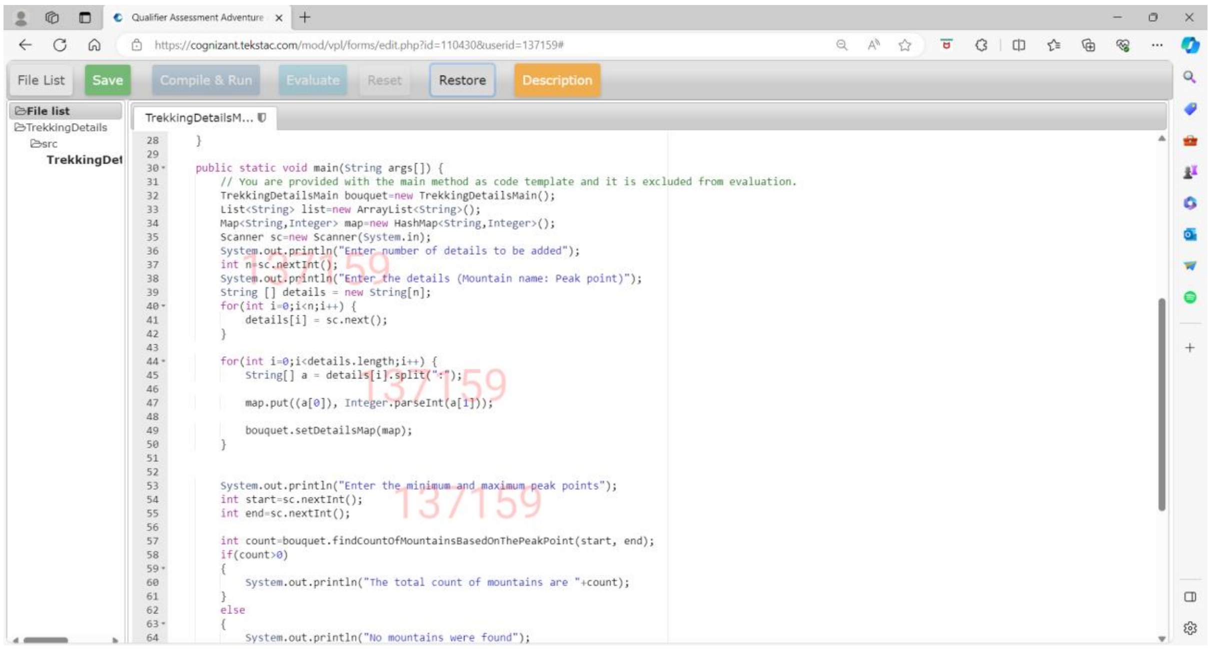
Enter the peak point to be searched

200

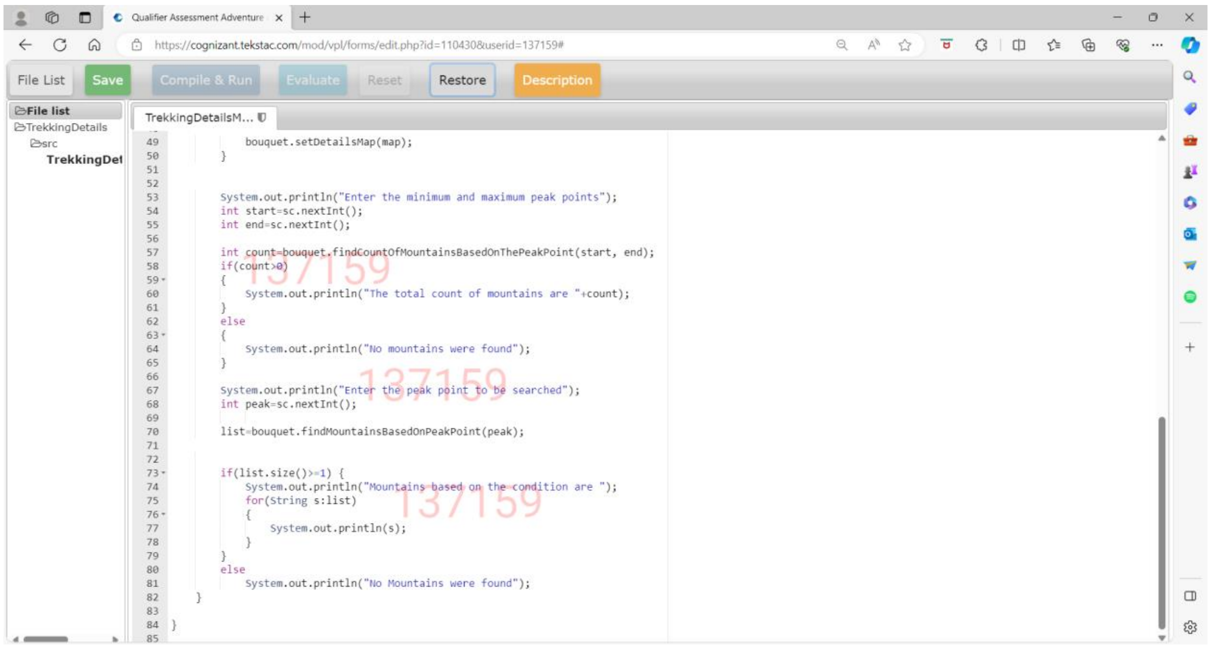
No Mountains were found



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

public class TrekkingDetailsMain {

    private Map<String,Integer> detailsMap=new HashMap<String,Integer>();

    public Map<String, Integer> getDetailsMap() {
        return detailsMap;
    }

    public void setDetailsMap(Map<String, Integer> detailsMap) {
        this.detailsMap = detailsMap;
    }

    public int findCountOfMountainsBasedOnThePeakPoint(int minimumPeak, int maximumPeak) {
        //Fill the code
        return 0;
    }

    public List<String> findMountainsBasedOnPeakPoint(int peakPoint){
        //Fill the code
        return null;

    }

    public static void main(String args[]) {
        // You are provided with the main method as code template and it is excluded from evaluation.
        TrekkingDetailsMain bouquet=new TrekkingDetailsMain();
        List<String> list=new ArrayList<String>();
        Map<String,Integer> map=new HashMap<String,Integer>();
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter number of details to be added");
        int n=sc.nextInt();
```



```java
    }

    public static void main(String args[]) {
        // You are provided with the main method as code template and it is excluded from evaluation.
        TrekkingDetailsMain bouquet=new TrekkingDetailsMain();
        List<String> list=new ArrayList<String>();
        Map<String,Integer> map=new HashMap<String,Integer>();
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter number of details to be added");
        int n=sc.nextInt();
        System.out.println("Enter the details (Mountain name: Peak point)");
        String [] details = new String[n];
        for(int i=0;i<n;i++) {
            details[i] = sc.next();
        }

        for(int i=0;i<details.length;i++) {
            String[] a = details[i].split(":");

            map.put((a[0]), Integer.parseInt(a[1]));

            bouquet.setDetailsMap(map);
        }


        System.out.println("Enter the minimum and maximum peak points");
        int start=sc.nextInt();
        int end=sc.nextInt();

        int count=bouquet.findCountOfMountainsBasedOnThePeakPoint(start, end);
        if(count>0)
        {
            System.out.println("The total count of mountains are "+count);
        }
        else
        {
            System.out.println("No mountains were found");
```



```java
            bouquet.setDetailsMap(map);
        }


        System.out.println("Enter the minimum and maximum peak points");
        int start=sc.nextInt();
        int end=sc.nextInt();

        int count=bouquet.findCountOfMountainsBasedOnThePeakPoint(start, end);
        if(count>0)
        {
            System.out.println("The total count of mountains are "+count);
        }
        else
        {
            System.out.println("No mountains were found");
        }

        System.out.println("Enter the peak point to be searched");
        int peak=sc.nextInt();

        list=bouquet.findMountainsBasedOnPeakPoint(peak);


        if(list.size()>=1) {
            System.out.println("Mountains based on the condition are ");
            for(String s:list)
            {
                System.out.println(s);
            }
        }
        else
            System.out.println("No Mountains were found");
    }

}
```