Rajalakshmi Engineering College

Name: RamRohith Muthusamy
Email: 240701419@rajalakshmi.edu.in
Roll no: 240701419
Phone: 8248706370
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Ram wants to evaluate the time required to break even on an investment based on initial costs, monthly profits, and monthly expenses. Write a program to calculate the break-even point in months and categorize the return on investment.

Compute the break-even point by using the formula: initial cost / (monthly profit - monthly expenses)Based on the break-even point, classify the return on investment into one of the following categories:Quick Return: If the break-even point is 3 months or fewer.Average Return: If the break-even point is between 4 and 12 months, inclusive.Long-term Return: If the break-even point exceeds 12 months.

Ram is new to programming, so he seeks your assistance in creating the program.

Note: monthly profit is always greater than monthly expenses.

Input Format

The first line of input consists of a double value representing the initial cost.

The second line consists of a double value representing the monthly profit.

The third line consists of a double value representing the monthly expenses.

Output Format

The first line prints "Break-even Point:", followed by the break-even point as a decimal number (of double datatype), formatted to two decimal places.

The second line prints "Category: ",followed by the investment return as a String, which can be one of:

- "Quick Return" if break-even point ≤ 3
- "Average Return" if break-even point ≤ 12
- "Long-term Return" if break-even point > 12

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10000.50
5000.75
1000.10

Output: Break-even Point: 2.50
Category: Quick Return

Answer

```
import java.util.Scanner;

public class Main{
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      double initialCost = scanner.nextDouble();
      double monthlyProfit = scanner.nextDouble();
```

```
        double monthlyExpenses = scanner.nextDouble();
        scanner.close();
        double breakEvenPoint = initialCost / (monthlyProfit - monthlyExpenses);
        String category;
        if (breakEvenPoint <= 3) {
            category = "Quick Return\n";
        } else if (breakEvenPoint <= 12) {
            category = "Average Return\n";
        } else {
            category = "Long-term Return\n";
        }
        System.out.printf("Break-even Point: %.2f Category: %s\n", breakEvenPoint,
category);
    }
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

John is a fitness trainer, and he wants to use the BMI calculator to assess the body mass index of his clients. He has a list of clients based on their height and weight.

John plans to write a program to quickly determine the BMI and provide a classification for each client.

If BMI is less than 18.5, the program will classify it as "Underweight"If BMI is between 18.6 and 24.9, the program will classify it as "Normal Weight"If BMI is between 25.0 and 29.9, the program will classify it as "Overweight"If BMI is 30.0 or higher, the program will classify it as "Obese"

Note: Formula to calculate BMI = weight/(height*height)

Input Format

The first line of input consists of a double value, representing the height of the person in meters.

The second line consists of a double value, representing the weight of the person in kilograms.

Output Format

The first line of output prints "BMI: " followed by a double (rounded to two decimal places) representing the calculated BMI.

The second line prints "Classification: " followed by a string indicating the BMI category (Underweight, Normal Weight, Overweight, or Obese).

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1.2
45.2

Output: BMI: 31.39
Classification: Obese

Answer

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double height = scanner.nextDouble();
        double weight = scanner.nextDouble();
        double bmi = weight / (height * height);
        bmi = Math.round(bmi * 100.0) / 100.0;
        String classification;
        if (bmi < 18.5) {
            classification = "Underweight";
        } else if (bmi >= 18.6 && bmi <= 24.9) {
            classification = "Normal Weight";
        } else if (bmi >= 25.0 && bmi <= 29.9) {
            classification = "Overweight";
        } else {
            classification = "Obese";
        }
        System.out.printf("BMI:%.2f " , bmi);
        System.out.println("\nClassification: " + classification);
        scanner.close();
    }
}
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Samantha is a diligent math student who is exploring the world of programming. She is learning Java and has recently studied conditional statements. One day, her teacher gives her an interesting problem to solve, which takes a number as input and checks whether it is a multiple of 5 or 7.

Help her complete the task.

Input Format

The input consists of a single integer N, representing the number to be checked.

Output Format

If the number is a multiple of 5 but not 7, the output prints "N is a multiple of 5".

If the number is a multiple of 7, the output prints "N is a multiple of 7".

Otherwise the output prints "N is neither multiple of 5 nor 7" where N is an entered integer.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10
Output: 10 is a multiple of 5

Answer

```
import java.util.Scanner;
import java.io.*;

class Main{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();

        if(a%5 == 0){
            System.out.println(a+" is a multiple of 5");
        }
        else if(a%7 == 0){
                    System.out.println(a+" is a multiple of 7");
        }
        else{
            System.out.println(a+" is neither multiple of 5 nor 7");
        }
    }
}
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Noah is analyzing numbers within a given range [A, B] and wants to calculate a special sum. For each number in the range, he calculates the product of its odd digits (ignoring even digits). If the number contains no odd digits, it is skipped. The sum of these products for all numbers in the range is the result.

Write a program to compute this sum.

Example

Input:

10 12

Output:

3

Explanation:

For 10, odd digits = 1, product = 1.

For 11, odd digits = 1, 1, product = 1 * 1 = 1.

For 12, odd digits = 1, product = 1.

Total sum = 1 + 1 + 1 = 3

Input Format

The input consists of two space-separated integers A and B, representing the inclusive range boundaries.

Output Format

The output prints a single integer representing the sum of the products of odd digits for all numbers in the range.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10 12
Output: 3

Answer

```
import java.util.Scanner;

public class Main {
   public static int productOfOddDigits(int num) {
      int product = 1;
      boolean hasOdd = false;
      while (num > 0) {
         int digit = num % 10;
         if (digit % 2 == 1) {
            product *= digit;
            hasOdd = true;
         }
         num /= 10;
      }
      return hasOdd ? product : 0;
   }
```

```
    public static int specialSum(int A, int B) {
        int total = 0;
        for (int i = A; i <= B; i++) {
            total += productOfOddDigits(i);
        }
        return total;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int A = scanner.nextInt();
        int B = scanner.nextInt();
        scanner.close();

        System.out.println(specialSum(A, B));
    }
}
```

Status : Correct ***Marks : 10/10***