Rajalakshmi Engineering College

Name: Lakshitha K
Email: 241801132@rajalakshmi.edu.in
Roll no: 241801132
Phone: 6381920328
Branch: REC
Department: AI & DS - Section 3
Batch: 2028
Degree: B.E - AI & DS

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

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

```
// You are using Java
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);
        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\n", bmi);
    System.out.println("Classification: " + classification);

    scanner.close();
    }
}
```

Status : Correct   *Marks : 10/10*

2.   Problem Statement

Maya, a student in an arts and crafts class, wants to create a pattern using stars (*) in a specific format. She plans to use a program to help her construct the pattern.

Write a program that takes an integer as input and constructs the following pattern using nested for loops.

Input: 5

Output:
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*

Input Format

The input consists of a number (integer) representing the number of rows.

Output Format

The output displays the required pattern.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5
Output: *
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

Answer

```java
// You are using Java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int rows = scanner.nextInt();
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
        }
        for (int i = rows - 1; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
        }
```

```
        scanner.close();
    }
}
```

Status : Correct   *Marks : 10/10*

3. Problem Statement

Raj is solving a physics problem involving projectile motion, where he
needs to calculate the time a ball hits the ground using a quadratic
equation of the form ax² + bx + c = 0. Depending on the coefficients, the
ball may hit the ground once, twice, or not at all in real time.

Help Raj find all real roots of the equation, if any.

Note: discriminant = b2 - 4ac

Input Format

The input consists of three space-separated doubles a, b, and c, representing the
coefficients of the quadratic equation.

Output Format

If there are two real roots, print:

- "Two real solutions:"
- "Root1 = <value>"
- "Root2 = <value>"

If there is one real root, print:

- "One real solution:"
- "Root = <value>"

If there are no real roots, print:

- "There are no real solutions."

Note: values are rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1 6 9
Output: One real solution:
Root = -3.00

Answer

```java
// You are using Java
import java.util.Scanner;

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

        // Read coefficients a, b, and c
        double a = scanner.nextDouble();
        double b = scanner.nextDouble();
        double c = scanner.nextDouble();

        // Calculate the discriminant
        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            // Two real solutions
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);

            // Format roots with two decimal places, smallest root as Root2
            double r1 = Math.max(root1, root2);
            double r2 = Math.min(root1, root2);

            System.out.printf("Two real solutions:\nRoot1 = %.2f\nRoot2 = %.2f\n", r1,
r2);
        } else if (discriminant == 0) {
            // One real solution
            double root = -b / (2 * a);
            System.out.printf("One real solution:\nRoot = %.2f\n", root);
        } else {
```

```
        // No real solutions
        System.out.println("There are no real solutions.");
    }

    scanner.close();
    }
}
```

Status : Correct   *Marks : 10/10*

4. Problem Statement

Joe has a favourite number, let's call it X. He wants to check if X is divisible by the sum of its digits. If it is, he considers it a lucky number. If not, he wants to find the closest smaller number, that is divisible by the sum of digits of X. Joe has challenged his friends to solve this puzzle at his birthday party.

Example

Input:

157

Output:

157 is not divisible by the sum of its digits.

The closest smaller number that is divisible: 156

Explanation:

The sum of the digits of X is 1+5+7=13. Since 157 is not divisible by 13, we need to find the closest smaller number that is divisible by 13. 156 is divisible by 13, it is the closest smaller number that meets the requirement.

Input Format

The input consists of an integer X, representing Joe's favourite number.

Output Format

If X is a lucky number, then the output must be in the format: "X is divisible by the sum of its digits."

If not, then the output must be in the format:

"X is not divisible by the sum of its digits.

The closest smaller number that is divisible: Y",

where X is the entered number and Y is the closest number.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 120
Output: 120 is divisible by the sum of its digits.

Answer

```java
// You are using Java
import java.util.Scanner;

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

        if (x % sumOfDigits == 0) {
            System.out.println(x + " is divisible by the sum of its digits.");
        } else {
            int closest = x - 1;
            while (closest > 0) {
                if (closest % sumOfDigits == 0) {
                    break;
                }
                closest--;
            }
            System.out.println(x + " is not divisible by the sum of its digits.");
            System.out.println("The closest smaller number that is divisible: " +
closest);
        }
```

```
        scanner.close();
    }
    private static int getSumOfDigits(int num) {
        int sum = 0;
        while (num != 0) {
            sum += num % 10;
            num /= 10;
        }
        return sum;
    }
}
```

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