Rajalakshmi Engineering College

Name: ANIRUDH R
Email: 240701039@rajalakshmi.edu.in
Roll no: 240701039
Phone: 9363540767
Branch: REC
Department: CSE - Section 10
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

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

```
import java.util.Scanner;

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

```
        int n = scanner.nextInt();
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
        for (int i = n - 1; i > 0; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

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

2. 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

```
import java.util.Scanner;

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

      int sumOfDigits = 0;
      int originalNumber = number;

      while (number > 0) {
         sumOfDigits += number % 10;
```

```
        number /= 10;
    }

    boolean isDivisible = originalNumber % sumOfDigits == 0;

    if (isDivisible) {
        System.out.println(originalNumber + " is divisible by the sum of its
digits.");
    } else {
        int closestSmallerNumber = -1;
        number = originalNumber - 1;

        while (number > 0) {
            if (number % sumOfDigits == 0) {
                closestSmallerNumber = number;
                break;
            }
            number--;
        }

        System.out.println(originalNumber + " is not divisible by the sum of its
digits.");
        if (closestSmallerNumber != -1) {
            System.out.println("The closest smaller number that is divisible: " +
closestSmallerNumber);
        }
    }
  }
}
```

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 $ax^2 + bx + c = 0$. 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

```
import java.util.Scanner;

class QuadraticEquationSolver {
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);

        double a = scanner.nextDouble();
        double b = scanner.nextDouble();
        double c = scanner.nextDouble();
        scanner.close();

        double discriminant = (b * b) - (4 * a * c);

        if (discriminant > 0) {
            double sqrtD = Math.sqrt(discriminant);

            double root1 = (-b + sqrtD) / (2 * a);
            double root2 = (-b - sqrtD) / (2 * a);

            System.out.printf("Two real solutions:%nRoot1 = %.2f%nRoot2 = %.2f%n",
root1, root2);
        }
        else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.printf("One real solution:%nRoot = %.2f%n", root);
        }
        else {
            System.out.println("There are no real solutions.");
        }
    }
}
```

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

4. Problem Statement

Ted, the computer science enthusiast, has accepted the challenge of writing a program that checks if the number of digits in an integer matches the sum of its digits.

Guide Ted in designing and writing the code to solve this problem using a 'do-while' loop.

Input Format

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

Output Format

If the sum is equal to the number of digits, print "The number of digits in N matches the sum of its digits."

Else, print "The number of digits in N does not match the sum of its digits."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20
Output: The number of digits in 20 matches the sum of its digits.

Answer

```
import java.util.Scanner;

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

        int digitCount = 0;
        int digitSum = 0;
        int tempNumber = Math.abs(number);

        do {
            int digit = tempNumber % 10;
            digitCount++;
            digitSum += digit;
            tempNumber /= 10;
        } while (tempNumber != 0);

        if (digitCount == digitSum) {
            System.out.println("The number of digits in " + number + " matches the
sum of its digits.");
        } else {
            System.out.println("The number of digits in " + number + " does not match
the sum of its digits.");
```

```
    }
  }
}
```

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