

# Rajalakshmi Engineering College

Name: Elamvazhuthi MS

Email: 240701133@rajalakshmi.edu.in

Roll no: 240701133

Phone: 7010282287

Branch: REC

Department: CSE - Section 4

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**

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**

```
// You are using Java
import java.util.*;
class m{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int c=0;
        int d=a;
        int b=0;
        do{
            int f=d%10;
            b=b+f;
            c++;
            d=d/10;
        }while(d>0);
        if(b==c){
            System.out.printf("the number of digits in %d matches the sum of its
digits.",a);
        }
        else{
            System.out.printf("the number of digits in %d does not match the sum
of its digits.",a);
        }
    }
}
```

**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**

```
// You are using Java
import java.util.*;
class s{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int c=a;
        int b=0;
        int k=c;
        while(a>0){
            int d=a%10;
            b=b+d;
            a=a/10;
        }
        if(c%b==0){
            System.out.printf("%d is divisible by the sum of its digits.",c);
        }
        else if(c%b!=0){
            //int k=c;
            while(c%b!=0){
                c=c-1;
            }
            System.out.printf("%d is not divisible by the sum of its digits.",k);
            System.out.printf("\nthe closest smaller number that is divisible: %d",c);
        }
    }
}
```

**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 =  $b^2 - 4ac$

#### ***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***

```

// You are using Java
import java.util.*;
class s{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        double a=s.nextDouble();
        double b=s.nextDouble();
        double c=s.nextDouble();
        double v=(b*b)-4*a*c;
        //double d=a*(v*v)+b*v+c;

        if(v<0){
            System.out.printf("there are no real solutions.");
        }
        else if(v==0) {
            double r=-b/(2*a);
            System.out.printf("one real solution:\n");
            System.out.printf("root = %.2f",r);
        }
        else if(v>0){
            double sq=Math.sqrt(v);
            double r1=(-b+sq)/(2*a);
            double r2=(-b-sq)/(2*a);
            System.out.printf("two real solutions:\n");
            System.out.printf("root1= %.2f\n",r1);
            System.out.printf("root2 = %.2f\n",r2);
        }
    }
}

```

**Status : Correct**

**Marks : 10/10**

#### 4. 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

```
// You are using Java
import java.util.*;
class s{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int i=0;
        for(i=1;i<=a;i++){
            for(int j=1;j<=i;j++){
                System.out.printf("* ");
            }
            System.out.printf("\n");
        }
        for(i=a-1;i>0;i--){
            for(int j=1;j<=i;j++){
                System.out.printf("* ");
            }
            System.out.printf("\n");
        }
    }
}
```

Status : Correct

Marks : 10/10