Rajalakshmi Engineering College

Name: SIVAGURU D
Email: 240701517@rajalakshmi.edu.in
Roll no: 240701517
Phone: 9345616842
Branch: REC
Department: CSE - Section 7
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.*;
class Main{
   public static void main(String args[])
   {
      Scanner s = new Scanner(System.in);
```

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

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.*;
public class Main {
   public static void main(String[] args) {
      Scanner s= new Scanner(System.in);
      int n = s.nextInt();
      int sum = 0;
      int o = n;
      while (n > 0) {
         sum += n % 10;
         n/= 10;
      }
      boolean i = o % sum == 0;

      if (i) {
         System.out.println(o + " is divisible by the sum of its digits.");
      } else {
         int c = -1;
         n = o - 1;

         while (n>0) {
             if (n%sum == 0) {
                 c = n;
                 break;
             }
             n--;
         }

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

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

3. 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.*;
class Main{
   public static void main(String args[])
   {
      Scanner s = new Scanner(System.in);
      double a=s.nextDouble(),b=s.nextDouble(),c=s.nextDouble();
      float bep=(float)(a/(b-c));
      System.out.printf("Break-even Point: %.2f\n",bep);
```

```
        if(bep<=3.0)
        System.out.println("Category: Quick Return");
        else if(bep>=4 && bep<=12)
        System.out.println("Category: Average Return");
        else
        System.out.println("Category: Long-term Return");
    }
}
```

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.*;
class Main{
   public static void main(String args[])
   {
      Scanner s = new Scanner(System.in);
      String a=s.nextLine();
      int b=a.length();
      int c=0;
      for(int i=0;i<b;i++)
      {
         c+=a.charAt(i)-'0';
      }
      if(b==c)
      System.out.println("The number of digits in "+a+"matches the sum of its
digits.");
      else
      System.out.println("The number of digits in "+a+" does not match the sum
of its digits.");
   }
}
```

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