Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arun is working on a project to automate the process of determining whether a student has passed or failed based on their subject marks.

He aims to create a simple program that takes positive integers as marks for five subjects from the user. If the average of the marks is greater than or equal to 50, the student has passed the exam. Otherwise, the student has failed.

Help Arun to implement the project.

Input Format

The input consists of five space-separated integers, representing the marks in five subjects.

Output Format

The first line of output prints "Average score: " followed by an integer representing the average score.

The second line prints one of the following:

1. If the condition is satisfied, print "The student has passed".
2. Otherwise, the output prints "The student has failed".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50 60 70 80 90

Output: Average score: 70
The student has passed

Answer

```
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int a=sc.nextInt();
      int b=sc.nextInt();
      int c=sc.nextInt();
      int d=sc.nextInt();
      int e=sc.nextInt();
      System.out.printf("Average score: %d",(a+b+c+d+e)/5);
      if ((a+b+c+d+e)/5 >=50 ){
         System.out.println("\nThe student has passed");
      }
      else{
         System.out.println("\nThe student has failed");
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. 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.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int n=sc.nextInt();
      if(n%5==0){
         System.out.printf("%d is a multiple of 5",n);
      }
      else if (n%7==0){
         System.out.printf("%d is a multiple of 7",n);
      }
      else{
         System.out.printf("%d is neither multiple of 5 nor 7",n);
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

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

```
import java.util.*;
class main{
    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);;
        float h=sc.nextFloat();
        float w=sc.nextFloat();
        double bmi = w/(h*h);
        System.out.printf("BMI: %.2f\n",bmi);
        System.out.println("Classification: ");
        if (bmi<18.5){
            System.out.println("Underweight");
        }
        else if (bmi>18.6 && bmi<24.9){
            System.out.println("Normal Weight");
        }
        else if (bmi<25.0 && bmi>29.9){
            System.out.println("Overweight");
```

```
        }
        else{
            System.out.println("Obese");
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 6.5

Section 1 : Coding

1. Problem Statement

Amit wants to evaluate the depreciation of his car over time to understand its current value and categorize it based on that value.

Write a program that helps him determine the current value of his car after a certain number of years of depreciation and classify it into one of three categories:

High: If the current value is greater than 10,000.Medium: If the current value is between 5,000 and 10,000, both inclusive.Low: If the current value is less than 5,000.

The depreciation rate of the car is 15% per year. The program should calculate the current value of the car after applying this depreciation over the given number of years and print the current value along with the category.

Input Format

The first line of input consists of an integer, representing the initial cost of the car.

The second line consists of an integer, representing the number of years the car has been depreciating.

Output Format

The first line of output prints a double value, representing the current value of the car, rounded off to two decimal places "Current Value: <value>".

The second line prints its category "Category: <categories>".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20000
5
Output: Current Value: 8874.11
Category: Medium

Answer

```
import java.util.Scanner;
class main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int initialCost = scanner.nextInt();
        int years = scanner.nextInt();
        double depreciationRate = 0.15;
        double currentValue = initialCost * Math.pow(1 - depreciationRate, years);
        currentValue = Math.round(currentValue * 100.0) / 100.0;
        System.out.println("Current Value: " + currentValue);
        if (currentValue > 10000) {
            System.out.println("Category: High");
        } else if (currentValue >= 5000) {
            System.out.println("Category: Medium");
        } else {
            System.out.println("Category: Low");
```

```
    }
  }
}
```

Status : Partially correct ***Marks : 6.5/10***

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

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

```
import java.util.*;
class main{
   public static void main(String args[]){
      Scanner sc = new Scanner(System.in);
      int n=sc.nextInt();
      int temp=n;
      int sum=0;
      int digit_count=0;
      do{
         int digit=temp%10;
         sum+=digit;
         digit_count++;
         temp/=10;
      }
      while (temp>0);
      if (digit_count==sum){
         System.out.printf("The number of digits in %d matches the sum of its
digits.",n);
      }
      else{
         System.out.printf("The number of digits in %d does not match the sum of
its digits.",n);
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q6

Attempt : 1
Total Mark : 10
Marks Obtained : 10

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 sc = new Scanner(System.in);
      int n=sc.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>=1;i--){
            for(int j=1;j<=i;j++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q7

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are taking part in a coding challenge where your task is to design a program that conjures a mesmerizing numerical pyramid pattern. The enchanting pattern is fashioned using a for loop and is customized based on user input.

Participants are prompted to unveil the pyramid's magic by specifying its height - essentially dictating the number of rows in this spellbinding creation.

Write a program that employs to weave this captivating numerical pyramid as shown below.

Example

Input:
4

Output:

Input Format

The input consists of a positive integer n representing the number of rows in the pattern.

Output Format

The output prints the required pyramid pattern, as shown in the sample output.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4

Output:
```
   1
  123
 12345
1234567
```

Answer

```
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int n=sc.nextInt();
      for(int i=1;i<=n;i++){
         int num_count=2*i-1;
         for(int j=1;j<=num_count;j++){
            System.out.print(j);
         }
         System.out.println();
      }
   }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q8

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A bank generates secure codes using 3-digit numbers where each digit is unique, and the code must be divisible by 3. You are tasked with generating the first N such codes based on user input, ensuring the digits are unique and the number is divisible by 3.

Note: Use nested for loops to solve.

Input Format

The first line contains an integer N representing the number of valid codes to generate.

Output Format

The output prints N line,s each line contains a valid 3-digit code.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

Output: 102
105
108
120
123

Answer

```
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int n=sc.nextInt();
      int count=0;
      for(int i=100;i<=999;i++){
         int a=i/100;
         int b=(i/10)%10;
         int c=i%10;
         if(a!=b && b!=c && a!=c){
            if(i%3==0){
               System.out.println(i);
               count++;
               if(count==n){
                  break;
               }
            }
         }
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_PAH

Attempt : 1
Total Mark : 40
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Ravi wants to estimate the total utility bill for a household based on the consumption of electricity, water, and gas.

Write a program to calculate the total bill using the following criteria:

The cost per unit for electricity is 0.12, for water is 0.05, and for gas is 0.08.A discount is applied to the total cost based on the following conditions:If the total cost is 100 or more, a 10% discount is applied.If the total cost is between 50 and 99.99, a 5% discount is applied.No discount is applied if the total cost is less than 50.

The program should output the total bill after applying the discount with two decimal places.

Input Format

The input consists of three double values, representing the number of units consumed for electricity, water, and gas respectively.

Output Format

The output prints a double value, representing the total bill after applying the discount, formatted to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1000.0
200.0
100.0
Output: 124.20

Answer

```
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      double e=sc.nextDouble();
      double w=sc.nextDouble();
      double g=sc.nextDouble();
      double ec=0.12;
      double wc=0.05;
      double gc=0.08;
      double total = (e*ec) + (w*wc) + (g*gc);
      if (total>=100){
         System.out.printf("%.2f",total-(total*10/100));
      }
      else if (total>=50 && total<=99.99){
         System.out.printf("%.2f",total-(total*5/100));
      }
      else{
         System.out.printf("%.2f",total);
      }
   }
}
```

Status : Correct *Marks : 10/10*

2. Problem Statement

Sampad is a high school teacher who wants to convert numeric grades into letter grades.

Write a program that accepts a numeric grade as input. The program should then convert this numeric grade into a letter grade based on specific conditions. The letter grades are A, B, C, D and F.

The conversion is determined by the following conditions:

If the numeric grade is 90 or higher, it's an "A"If the numeric grade is between 80 and 89 (inclusive), it's a "B"If the numeric grade is between 70 and 79 (inclusive), it's a "C"If the numeric grade is between 60 and 69 (inclusive), it's a "D"If the numeric grade is below 60, it's an "F"

Input Format

The input consists of an integer representing the numeric grade of the student.

Output Format

The output prints the letter grade corresponding to the input numeric grade as "Letter Grade: <grade>".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 85
Output: Letter Grade: B

Answer

```
import java.util.*;
class main{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int grade=sc.nextInt();
```

```
        if (grade>=90){
            System.out.println("Letter Grade: A");
        }
        else if (grade>=80 && grade<=89){
            System.out.println("Letter Grade: B");
        }
        else if (grade>=70 && grade<=79){
            System.out.println("Letter Grade: C");
        }
        else if (grade>=60 && grade<=69){
            System.out.println("Letter Grade: D");
        }
        else{
            System.out.println("Letter Grade: F");
        }
    }
}
```

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

3. Problem Statement

Rohit is tasked with designing a program to analyze the digits of a given integer.

Write a program to help Rohit that takes an integer as input and identifies the minimum odd digit and the maximum even digit present in the number. If no odd or even digits are present, display appropriate messages.

Implement the solution using a 'while' loop to iterate through the digits of the given number.

Input Format

The input consists of an integer n.

Output Format

The first line of output prints the message "Minimum odd digit: " followed by an integer representing the smallest odd digit found in the input number.

If no odd digit exists, it prints "There are no odd digits in the number."

The second line of output prints the message "Maximum even digit: " followed by an integer representing the largest even digit found in the input number.

If no even digit exists, it prints "There are no even digits in the number."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3465
Output: Minimum odd digit: 3
Maximum even digit: 6

Answer

```
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int n=sc.nextInt();
      int minOdd=10;
      int maxEven=-1;
      boolean hasOdd = false;
      boolean hasEven = false;
      while (n>0){
         int digit=n%10;
         if(digit%2==0){
            hasEven=true;
            if(digit>maxEven){
               maxEven=digit;
            }
         }else{
            hasOdd=true;
            if(digit<minOdd){
               minOdd=digit;
            }
         }
         n=n/10;
      }
      if (hasOdd){
```

```
            System.out.printf("Minimum odd digit: %d\n",minOdd);
        }
        else{
            System.out.printf("There are no odd digits in the number.\n");
        }
        if(hasEven){
            System.out.printf("Maximum even digit: %d\n",maxEven);
        }
        else{
            System.out.println("There are no even digits in the number.\n");
        }
    }
}
```

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

4. Problem Statement

You are given a number of distribution centers (rows) and are tasked with generating a zigzag shipment route pattern. Each shipment route should alternate between left-to-right and right-to-left, as described below.

The program should print the zigzag pattern with a tab (\t) separating the columns. For each row, the shipment numbers should follow a diagonal pattern where numbers are placed in a zigzag, left to right on odd rows and right to left on even rows.

Input Format

The input consists of an integer N, which represents the number of distribution centers (rows) for the zigzag pattern.

Output Format

The output prints the zigzag pattern with N rows, formatted with a tab space (\t) separating the columns.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

Output:

```
        1
     2    6
   3    7    10
 4    8    11    13
5    9    12    14    15
```

Answer

```
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int n=sc.nextInt();
      int num=1;
      for(int i=1;i<=n;i++){
         for(int t=0;t<n-i;t++){
            System.out.println("\t");
         }
         int[] row=new int[i];
         for(int j=0;j<i;j++){
            row[j]=num++;
         }
         if(i%2==0){
            for(int j=i-1;j>=0;j--){
               System.out.print(row[j] + "\t");
            }
         }else{
            for(int j=0;j<i;j++){
               System.out.print(row[j] + "\t");
            }
         }
         System.out.println();
      }
   }
}
```

Status : Wrong ***Marks : 0/10***

Rajalakshmi Engineering College

Name: Mohamed Humaid Zahir Hussain
Email: 240701322@rajalakshmi.edu.in
Roll no: 2116240701322
Phone: 6382261139
Branch: REC
Department: CSE - Section 9
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 : 36

Section 1 : Coding

1. 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.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      double a=sc.nextDouble();
      double b=sc.nextDouble();
      double c=sc.nextDouble();
      double discriminant=b*b-4*a*c;
      if (discriminant>0){
         double root1=(-b+Math.sqrt(discriminant))/(2*a);
         double root2=(-b-Math.sqrt(discriminant))/(2*a);
         System.out.println("Two real solutions:");
         System.out.printf("Root1 = %.2f\n",root1);
         System.out.printf("Root2 = %.2f\n",root2);
```

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

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.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int x=sc.nextInt();
      int temp=x,sum=0;
      while(temp>0){
         sum+=temp%10;
         temp/=10;
      }
      if(x%sum==0){
         System.out.printf("%d is divisible by the sum of its digits.",x);
      }
      else{
         System.out.printf("%d is not divisible by the sum of its digits.\n",x);
         int closest=x-1;
         while(closest>0){
            int t=closest,s=0;
            while(t>0){
```

```
            s+=t%10;
            t/=10;
        }
        if(closest%s==0){
            break;
        }
        closest--;
    }
    System.out.printf("The closest smaller number that is divisible:
%d",closest);
  }
 }
}
```

Status : Partially correct ***Marks : 6/10***

3. 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 sc = new Scanner(System.in);
      int n=sc.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>=1;i--){
         for(int j=1;j<=i;j++){
```

```
            System.out.print("*");
        }
        System.out.println();
    }
  }
}
```

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

4. 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.*;
class main{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      double h=sc.nextDouble();
      double w=sc.nextDouble();
      double bmi=w/(h*h);
      System.out.printf("BMI: %.2f",bmi);
      if (bmi<18.5){
         System.out.println("\nClassification: Underweight");
      }
      else if (bmi>18.6 && bmi<24.9){
         System.out.println("\nClassification: Normal Weight");
      }
      else if (bmi>25.0 && bmi<29.9){
         System.out.println("\nClassification: Overweight");
      }
      else{
         System.out.println("\nClassification: Obese");
      }
   }
}
```

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