```
/** Q1
         * Write a program to read the radius of a circle and calculate
         * its area using the formula:
                 A = π * r²

        Problem Flow:
        1. User inputs radius (r)
        2. Program calculates: area = (22.0 / 7) * r * r
        3. Output displays area with radius

        Input / Output Example:
        Input:
        Enter Radius of Circle: 7
        Output:
        Area of Circle with Radius 7.0000 is 154.0000
*/

/** Q2
         * Write a Java program to calculate Simple Interest using the formula:
                 SI = (P × N × R) / 100
                 Where:
                 P = Principal amount
                 N = Number of years
                 R = Rate of interest

                 Problem Flow:
                 User inputs:
                         Principal (P)
                         Rate (R)
                         Years (N)
                 Program calculates:
```

```
            SI = (P × N × R) / 100
        Output displays the Simple Interest value.

        Input / Output Example:
        Input:
        Enter Principal Amount: 10000
        Enter Rate of Interest: 5
        Enter No. of Years: 2
        Output:
        Simple Interest = 1000.00
*/

/** Q3
    * This program calculates Compound Interest using the formula:
        $F = P \times (1 + i)^n$
        Where:
            P = Principal amount
            i = Rate/100
            n = Number of years
            Here, F is the final amount (principal + interest)

         Problem Flow:
        Input P, R, and N from user
        Calculate i = R / 100
        Use formula F = P × (1 + i)^N
        Display the final compound amount

        Input / Output Example:
        Input:
        Enter Principal Value: 10000
        Enter Rate of Interest: 10
```

```
                Enter No. of Years: 2

                Output:

                Compound Interest 12100.00

*/


/** Q4

         * This program converts temperature from Fahrenheit to Celsius using the formula:

                C = (5/9) × (F - 32)


        Problem Flow:

        Input temperature in Fahrenheit

        Apply formula: C = (5 / 9.0) × (F - 32)

        Display result in Celsius


        Input / Output Example:

        Sample Inputs to Test (from question):

        68, 150, 212, 0, -22

        Sample Output (for 68°F):

        68.000000 deg F is 20.000000 deg C


         */


/** Q5

         *  This program calculates the Volume and Surface Area of a sphere

         *  using the following formulas:

                        Volume (V) = (4/3) × π × r³

                        Area (A) = 4 × π × r²


        Problem Flow:

        Get r (radius) from user

        Calculate area using A = 4πr²
        Calculate volume using V = (4/3)πr³
        Display both values

         Input / Output Example:
        Input:
        Enter Radius of Sphere: 7
        Output:
        Volume of Sphere: 1436.571429
        Area of Sphere  : 615.428571

*/

/** Q6
         * This program allows the user to enter 3 test marks of a student,
         calculates the total and average, and displays the result.

        Problem Flow:
        Get 3 marks from user
        Add marks → total
        Divide total by 3.0 → average
        Display total and average

        Input / Output Example:
        Input:
        Enter Mark 1: 80
        Enter Mark 2: 75
        Enter Mark 3: 85
        Output:
        Total Marks   = 240
        Average Marks = 80.0
         */
```

```
/** Q7
        * This program reads a five-letter word from the user and encodes
        it by subtracting 1 from the ASCII value of each character.
        It then prints the newly encoded word.

        Problem Flow:
       Get 5-letter word input
       Loop each character → subtract 1 from ASCII
       Build new string with changed characters
       Print encoded result

       Input / Output Example:
       Input:
       Enter a Five letter word: white
       Output:
       Encoded Word: vghsd
        */

/** Q8
        * This program calculates the sum of the first n odd integers
        (Example: 1 + 3 + 5 + ... + (2n - 1)).

        Problem Flow:
       Input n from user
       Loop from 1 to (2n−1) in steps of 2
       Accumulate the sum
       Print the result

       Input / Output Example:
       Input:

       Enter how many odd numbers to sum: 5

       Output:

       Sum of first 5 odd numbers is 25

       (1 + 3 + 5 + 7 + 9 = 25)


        */


/** Q9

        * This program finds the sum of even numbers between 2 and 30 (inclusive).


        Problem Flow:

       Start loop from 2 to 30

       Check if number is even (i % 2 == 0)

       Add to sum

       Print final sum


        Input / Output Example:

       Output:

       Sum of even numbers from 2 to 30 = 240

       (Even numbers: 2 + 4 + 6 + ... + 30 = 240)

        */


/** Q10

        * This program calculates and prints the product of odd numbers from 1 to 15.

               (Odd numbers: 1, 3, 5, 7, ..., 15)


               Problem Flow:

               Start from 1 to 15

               Increment by 2 to get odd numbers

               Multiply all odd numbers
        Print the final product

        Input / Output Example:
        Output:
        Product of odd numbers from 1 to 15 = 2027025
        (Odd numbers: 1 × 3 × 5 × 7 × 9 × 11 × 13 × 15 = 2027025)

    */

/** Q11
    * This program reads a number n from the user and prints the
    factorial of numbers from 1 to n in a table format.

    Problem Flow:
    Read n from user
    Start fact = 1
    Loop from 1 to n
    Multiply fact *= i
    Print each number with its factorial

    Sample Output:
    Input:
    Enter the N value: 5

    Output:
    N   |   Factorial
    --------------------
    1  |          1
    2  |          2
    3  |          6
    4  |         24
```

```
        5  |        120
         */


    /** Q12
         * This program reads 5 numbers (each between 1 and 30) from the user.
         For each number, it prints a line with that many asterisks *,
         forming a simple bar chart or histogram.

         Problem Flow:
        Read 5 numbers from the user
        Loop through each number
        For each number:
        Print the number
        Print that many * using inner loop

        Sample Output:
        Input:
        Enter 5 Numbers in a same line:
        3 7 2 6 1
        Output:
        3 ***
        7 *******
        2 **
        6 ******
        1 *

         */

   /** Q13
         * This program first reads how many numbers the user wants to enter
         (N). Then it reads those N numbers and finds the smallest among
         them.

         Problem Flow:
        Get total number of inputs N
        Read first number as smallest
        Loop remaining N-1 times
        Compare each number with smallest
        Update smallest if a smaller number is found
        Print final smallest number

        Sample Input / Output:
        Input:
        Enter how many numbers you want to input: 5
        Enter number 1: 22
        Enter number 2: 18
        Enter number 3: 31
        Enter number 4: 6
        Enter number 5: 27
        Output:
        The smallest number is: 6

         */

   /** Q14
         *  This program will read a line of text from the user, analyze
         each character, and count how many are letters, digits, whitespace,
         or other types of characters (like punctuation).

         Problem Flow:
        Prompt the user to input a line of text.
```

```
        Initialize counters for letters, digits, spaces, and other characters.

        Loop through each character and classify it into one of the categories.

        Display the count for each category: letters, digits, spaces, and others.


        Sample Input / Output:

        Input:

        Enter the text below:

        Hello World! 123


        Output:

        Letters : 10

        Digits : 3

        Space Chars : 2

        Others : 2


         */


/** Q15

         *  This program uses a switch statement to examine the value of

         an integer called flag. Based on its value, it prints one of the

         following messages:

                HOT if the flag has the value 1.

                LUKE WARM if the flag has the value 2.

                COLD if the flag has the value 3.

                OUT OF RANGE if the flag has any other value.


         Problem Flow:

        Prompt the user to input an integer value for the flag.

        Use a switch statement to check the value of the flag and assign the correct heat status.

        Print the flag value and its corresponding heat status.
```

```
	Sample Input / Output:
	Input:
	Enter Heat code (1-3): 2

	Output:
	Flag 2 is LUKE WARM



	 */

/** Q16
	 *  This program uses a switch statement to examine the value of a
	 char-type variable called colour and prints one of the following
	 messages, depending on the character assigned:
		RED if either r or R is assigned to colour.
		GREEN if either g or G is assigned to colour.
		BLUE if either b or B is assigned to colour.
		BLACK if any other character is assigned to colour.

		Problem Flow:
		Prompt the user to enter a single character for the color code.
		Use a switch statement to check the character and assign the corresponding color.
		Print the entered color code and the corresponding color name.

		Sample Input / Output:
		Input:
		Enter Color code character: g
		Output:
		Color Code g is GREEN

	 */
```

```
/** Q17
        *  This program checks the temperature value (temp) and determines
        the physical state of water:
                If temp < 0 → "ICE"
                If temp is between 0 and 100 (inclusive) → "WATER"
                If temp > 100 → "STEAM"

                Flow Summary:
                Get user input for temperature
                Use if-else to decide the state of water
                Display the result to the user

                Sample I/O:
                Input:
                Enter the water temperature: 120
                Output:
                Water status is STEAM for the Temperature 120.00

        */
```

```
/** Q18
        *  This program takes a string (text) and creates its reverse
        version in another string (reverseText). Each character from the
        original string is added in reverse order to the new string.

        Flow Summary
       Read the input string
       Loop from last character to first
       Use comma operator in loop (i--, j++)
```

```
        Add each character to reverseText
        Print the reversed output

        Sample I/O
        Input:
        Enter the string below:
        hello
        Output:
        Reversed Text: olleh

         */

/** Q19
         * This program reads a set of numbers from the user and calculates
         the cumulative product, which means multiplying all the numbers
         together one by one.

         Code Flow
        Get number count n
        Use for loop to input numbers
        Multiply each number with product
        Print final result

        Test Data
        Input:
        6.2, 12.3, 5.0, 18.8, 7.1, 12.8
        Output:
        Cumulative Total = 514537.165312


         */
```

```
/** Q20
     * This program calculates and displays the first n numbers in the
     Fibonacci sequence. Each number is the sum of the previous
     two numbers.

     Code Flow
    Input: number of terms n
    Initialize first two terms: f1 = 1, f2 = 1
    Loop from i = 3 to n:
          Find next term using f = f1 + f2
          Shift values for next round
    Display all terms

    Sample Input/Output
    Input:
    How many Fibonacci terms? 10
    Output:
    1 1 2 3 5 8 13 21 34 55

     */
```

```
/** Q21
     * This program reads a number n and prints the first n prime numbers.
     A prime number is a number that is divisible only by 1 and itself.

     Problem Flow
    Get input n (how many primes).
    Start checking from 2.
    Use isPrime() to validate.
```

```
        If prime, print it and increment count.

        Stop when count reaches n.


        Input / Output Example

        Input:

        How many Prime numbers? 10

        Output:

        2 3 5 7 11 13 17 19 23 29


         */


/**Q22

This program prints a right-aligned triangle star pattern using

         nested for loops. The number of lines (n) is provided by the user.


        Problem Flow

        Get number of lines from user.

        Loop from 1 to n (line-by-line).

        For each line:

        Print n - i spaces.

        Print i stars.

        Go to the next line.


        Input / Output Example

        Input:

        How many Lines? 5

        Output:

  *

 **

 ***

****
```

```
*****

            */

  /** Q23
  This program prints a numeric pyramid pattern where numbers
            increase from the middle outwards and then decrease symmetrically.
            The pattern is centered and uses nested loops

            Problem Flow Summary
           Ask user for number of lines (n).
           For each line i from 1 to n:
           Print spaces to align the pyramid.
           Print increasing numbers (i to i + i - 1).
           Print decreasing numbers back to i.
           Move to the next line after printing each row.

Input / Output Example
Input:
Enter Number of Lines: 5

Output:
    1
   232
  34543
 4567654
567898765



            */
     /** Q24
            * This program reads n numbers into an array, calculates the average
            of those numbers using a loop, and displays the result.
            It demonstrates how to use arrays and object-oriented concepts
            like classes and methods in Java.

           Problem Flow
           Read input size n
           Read n elements into array
           Loop through array to compute sum
           Divide sum by n to get average
           Display result

            Input / Output Example
Input:
Enter how many numbers: 5
Enter the numbers:
10
20
30
40
50

Output:
Average of the numbers: 30.00


            */
```

```
/** Q25
        * This program reads n integers into an array and finds the second
        largest number among them. It uses a class with methods for input,
        processing (finding second largest), and output. It showcases
        array handling and object-oriented principles in Java.

        Problem Flow
        Read input size n
        Store values in array
        Use loop to compare and find:
                largest
                secondLargest
        Display result

        Input / Output Example
Input:
Enter how many numbers: 6
Enter the numbers:
10
45
22
45
8
19

Output:
Second largest number is: 22


        */
```