Q1.

**Problem Statement**

You have been assigned the responsibility of implementing the **Area** class for the construction company's software application.

You have declared the following inline functions within the Area class: get(), squareArea(), rectangleArea(), circleArea(), and parallelogramArea(). These functions are used to calculate the area of different geometric shapes.

Include the necessary code for the class definition and method implementations. Then, create an object of the Area class in the main method and demonstrate how the methods can be called to calculate the areas of squares, rectangles, circles, and parallelograms. Provide the complete code solution, including the class definition, method implementations, and the main method where the object is created and the methods are called.

**Note:**This is a sample question asked in a HCL interview.

**Input Format**

The first value is an integer value for the side of a square.

The next two values are integer values for the length and breadth of a rectangle, respectively.

The fourth value is a floating-point value for the radius of a circle.

The last two values are floating-point values for the base and height of a parallelogram, respectively.

**Output Format**

The program should output the areas of the square, rectangle, circle, and parallelogram in separate lines with the corresponding labels.

The output should be rounded off to two decimal places for the area of the circle.

**Refer to the sample output for formatting specifications.**

**Constraints**

Inputs > 0

Take pi = 3.14

**Sample InputSample Output**

4

5 4

6.5

6.2 4.5

Area of the square: 16

Area of the rectangle: 20

Area of the circle: 132.66

Area of the parallelogram: 27.90

**Sample InputSample Output**

6

10 12

8

15.2 18.7

Area of the square: 36

Area of the rectangle: 120

Area of the circle: 200.96

Area of the parallelogram: 284.24

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2.

**Problem Statement**

You have been assigned a task to create a program that performs arithmetic operations using the **Operation** class and using inline functions.

You have declared the following inline functions within the class: get(), sum(), difference(), product(), division(), and modulus(). These functions are used to perform various arithmetic operations. Implement the class and its methods as described above.

Then, create an object of the Operation class and demonstrate how the methods can be called to perform the necessary arithmetic calculations. Provide the complete code solution for the program, including the class definition, method implementations, and the main method where the object is created and the methods are called.

**Note:** This is a sample question asked in a TCS interview.

**Input Format**

The input consists of two integers, separated by space.

**Output Format**

The output prints the result of arithmetic operations.

**Refer to the sample output for the formatting specifications.**

**Constraints**

All inputs should be integers only.

**Sample InputSample Output**

50 8

Addition of two numbers: 58

Difference of two numbers: 42

Product of two numbers: 400

Division of two numbers: 6

Modulus of two numbers: 2

**Sample InputSample Output**

120 188

Addition of two numbers: 308

Difference of two numbers: -68

Product of two numbers: 22560

Division of two numbers: 0

Modulus of two numbers: 120

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3.

**Problem Statement**

Suppose you are given a task to write a program that checks if a given number is an Armstrong number or not. Additionally, if the number is an Armstrong number, the program should print its factors. Write a code for the same using inline functions.

**Function specifications:**

**inline bool isArmstrong(int number**) - This calculates whether the given number is an Armstrong number or not.

**inline void printFactors(int number)** - This prints the factors of the given number.

**Note:** This is a sample question that can be asked in a TCS recruitment.

**Input Format**

The input consists of an integer, n.

**Output Format**

The output displays whether the given number is an Armstrong number or not.

Additionally, if the given number is an Armstrong number, the second line of output displays the factors of the given number separated by space.

**Refer to the sample outputs for the formatting specifications.**

**Constraints**

0 <= n <= 105

**Sample InputSample Output**

409

409 is not an Armstrong number.

**Sample InputSample Output**

153

153 is an Armstrong number.

1 3 9 17 51 153

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4.

**Problem Statement**

You are working on a software application that deals with different number formats. The application needs to display a given number in decimal, octal, and hexadecimal formats based on user input.

Write a program that takes an integer input from the user and displays the number in decimal, octal, and hexadecimal formats using **manipulator functions.**

**Note:**This is a sample question asked in a TCS interview.

**Input Format**

The input consists of an integer **N**.

**Output Format**

The first line of output prints the given integer in decimal format.

The second line of output prints the given integer in octal format.

The third line of output prints the given integer in hexadecimal format.

**Refer to the sample output for formatting specifications.**

**Constraints**

0 < N <= 2000

**Sample InputSample Output**

100

Decimal format: 100

Octal format: 144

Hexadecimal format: 64

**Sample InputSample Output**

126

Decimal format: 126

Octal format: 176

Hexadecimal format: 7e

**Sample InputSample Output**

191

Decimal format: 191

Octal format: 277

Hexadecimal format: bf

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5.

**Problem Statement**

You are developing a messaging application that encrypts messages by reversing them. As part of the encryption process, you need to display the reversed message character by character using manipulator functions.

Write a program that takes a message input from the user, reverse and displays it using the **manipulator function.**

**Note:** This is a sample question asked in a TCS interview.

**Input Format**

The input consists of the message as a string.

**Output Format**

The output prints the message that is reversed character by character.

**Sample InputSample Output**

Hello world!

!dlrow olleH

**Sample InputSample Output**

Programming is fun

nuf si gnimmargorP

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6.

**Problem Statement**

**﻿**Preethi is learning programming and she got interested in how inline functions work. So she has been practicing questions on that.

Help her solve the code for the following question statement: Create an inline function **isLeapYear()** to check if a given year is a leap year.

**Note:**This is a sample question asked in a Capgemini interview.

**Input Format**

The input consists of the year as an integer.

**Output Format**

The output displays whether the given year is a leap year or not.

**Refer to the sample output for the formatting specifications.**

**Constraints**

Year > 0

**Sample InputSample Output**

2000

2000 is a leap year.

**Sample InputSample Output**

1700

1700 is not a leap year.

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q7.

**Problem Statement**

**﻿**Matt is learning C++ and he got interested in how inline functions work. So he has been practicing questions on that. Help him solve the code for the following question statement: Write an inline function **countEvenNumbers()** to count the number of even numbers present after squaring the given number.

**Example 1**

**Input:**

12

**Output:**

2

**Explanation:**

Given number = 12. After squaring (12\*12 = 144). The number of even numbers present is 2.

**Example 2**

**Input:**

3

**Output:**

0

**Explanation:**

Given number = 3. After squaring (3\*3 = 9). The number of even numbers present is 0.

**Note:** This is a sample question that can be asked in a mPhasis recruitment.

**Input Format**

The input consists of a positive integer N.

**Output Format**

The output prints the count of the number of even numbers that are present in the square of the given number.

**Constraints**

0 <= N <= 1000

**Sample InputSample Output**

12

2

**Sample InputSample Output**

3

0

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q8.

**Problem Statement**

You are building a chatbot application that interacts with users through text messages. As part of the chatbot's functionality, you need to convert any sentence provided by the user into uppercase.

Write a program that takes a sentence input from the user, converts it to uppercase using the **manipulator function**, and displays the modified sentence as the chatbot's response.

**Note:** This is a sample question asked in an AMCAT interview.

**Input Format**

The input consists of the sentence that is given by the user.

**Output Format**

The output prints the modified uppercase sentence.

**Sample InputSample Output**

Hello World

HELLO WORLD

**Sample InputSample Output**

Programming is Fun.

PROGRAMMING IS FUN.

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q9.

**Problem Statement**

You are tasked with creating a program to analyze a series of test scores and provide statistical information. Write a program that calculates and displays the average, minimum, and maximum values of a given set of test scores using **manipulator functions.**

The average, minimum, and maximum values should be displayed with two decimal places. The program should use manipulator functions to achieve the desired formatting.

**Note:**This is a sample question asked in a Cocubes interview.

**Input Format**

The first line of input consists of the number of test scores **N**.

The second line of input consists of the **N**test scores, separated by space.

**Output Format**

The first line of output prints the average value, rounded off to two decimal places.

The second line of output prints the minimum value, rounded off to two decimal places.

The last line of output prints the maximum value, rounded off to two decimal places.

**Refer to the sample outputs for the formatting specifications.**

**Constraints**

0 < N <= 100

**Sample InputSample Output**

5

1 2 3 4 5

Average: 3.00

Minimum Value: 1.00

Maximum Value: 5.00

**Sample InputSample Output**

8

74 58 96 32 14 56 91 23

Average: 55.50

Minimum Value: 14.00

Maximum Value: 96.00

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q10.

**Problem Statement**

You are tasked with developing a program for a Math competition. One of the challenges in the competition requires participants to find the sum of the squares of the first n even numbers and odd numbers.

Write a C++ program that uses an inline function to calculate the sum of the squares of the first n even and odd numbers. The program should take an input value of n from the user and output the result. You need to implement the necessary functions and ensure that the program handles the computation accurately.

**Function specifications:**

**inline int sumOfSquaresEven** - calculates the sum of squares of the first n even numbers

**inline int sumOfSquaresOdd** - calculates the sum of squares of the first n odd numbers

**Note:**This is a sample question asked in Accenture recruitment.

**Input Format**

The input consists of a positive integer N.

**Output Format**

The first line of output displays the sum of squares of the first N even numbers.

The second line of output displays the sum of squares of the first N odd numbers.

**Constraints**

1 <= N <= 30

**Sample InputSample Output**

3

56

35

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q11.

**Problem Statement**

You are working on a scientific research project that involves analyzing large sets of experimental data. As part of your analysis, you need to display the numerical values in scientific notation for better readability and consistency.

Write a program that takes input data, performs calculations, and outputs the results in scientific notation using the **std::scientific**manipulator.

**Note:**This kind of question will be asked in Cocubes and AMCAT tests.

**Input Format**

The input consists of a number.

**Output Format**

The output prints the given number in scientific notation, using the manipulator.

**Sample InputSample Output**

123.45

1.234500e+02

**Sample InputSample Output**

0.001

1.000000e-03

**Sample InputSample Output**

-987654.321

-9.876543e+05

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q12.

**Problem Statement**

You are building a digital circuit simulation program where you need to manipulate and analyze integer values. The program should allow users to input an integer and perform various operations on it and it should print the binary representation, complement, left shift, and right shift, check if it is negative, and its absolute value. Refer the below explanations for each.

**Binary representation -** Binary representation is a way to express a number using only 0s and 1s.

**Complement -**Complement refers to flipping the bits of a binary number, changing 0s to 1s and 1s to 0s.

**Left shift -**Left shift is an operation that moves the bits of a binary number to the left, effectively multiplying it by a power of 2.

**Right shift -** Right shift is an operation that moves the bits of a binary number to the right, effectively dividing it by a power of 2.

**Check if it is negative -** Checking if it is negative determines if the number is less than zero.

**Absolute value -** Absolute value refers to the magnitude of a number without considering its sign.

Write a program for the same and print everything in a detailed manner using manipulator functions.

**Note:**This question can be asked in Cocubes and AMCAT recruitments.

**Input Format**

The input consists of an integer, N.

**Output Format**

The first line of output prints the binary representation of the given number.

The second line of output prints the complement of the given number.

The third line of output prints the left shift value of the given number.

The fourth line of output prints the right shift value of the given number.

The fifth line of output prints whether the givennumber is a negative number or not.

The last line of output prints the absolute value of the given number.

**Refer to the sample outputs for the formatting specifications.**

**Constraints**

-1000 <= N <= 1000

**Sample InputSample Output**

10

Binary representation: 00001010

Complement: -11

Left shift by 2: 40

Right shift by 2: 2

Is negative? false

Absolute value: 10

**Sample InputSample Output**

-5

Binary representation: 11111011

Complement: 4

Left shift by 2: -20

Right shift by 2: -2

Is negative? true

Absolute value: 5

Time Limit: - ms Memory Limit: - kb Code Size: - kb