Applied Coding - Language Skill (Repetitions-Simple Loops)

Solve the following problems using computer with help of Python/C++/Java/C# language as means of communication.

Problem 1: Sum of Integers

Given the value of n, write a program that displays the sum of first n integers.

Problem 2: Vowel and Consonant Count

Write a program that will examine each character in a given string and determine how many of the characters are vowels and how many are consonants.

Problem 3: Sum of the Digits

Write a program that reads an integer from the user and displays the sum of its digits. For example, if the user enters 31415 then your program should display 3+1+4+1+5=14.

Problem 4: Fizz-Buzz

Fizz-Buzz is a game that is sometimes played by children to help them learn about division. The players are commonly arranged in a circle so that the game can progress from player to player continually. The starting player begins by saying one, and then play passes to the player to the left. Each subsequent player is responsible for the next integer in sequence before play passes to the following player. On a player's turn they must either say their number or one of following substitutions:

• If the player's number is divisible by 3 then the player says fizz instead of their number.

• If the player's number is divisible by 5 then the player says buzz instead of their number.

A player must say both fizz and buzz for numbers that are divisible by both 3 and 5. Any player that fails to perform the correct substitution or hesitates before answering is eliminated from the game. The last player remaining is the winner. Write a program that displays the answers for the first 100 numbers in the FizzBuzz game. Each answer should be displayed on its own line.

Problem 5: Sum of Odd Integers

Given the value of n, write a program that displays the sum of first n odd integers. For given value of 3, the program should display 1+3+5=9.

Problem 6: Caesar Cipher

One of the first known examples of encryption was used by Julius Caesar. Caesar needed to provide written instructions to his generals, but he didn't want his enemies to learn his

plans if the message slipped into their hands. As a result, he developed what later became known as the Caesar cipher.
The idea behind this cipher is simple (and as such, it provides no protection against modern code breaking techniques). Each letter in the original message is shifted by 3 places. As a result, A becomes D, B becomes E, C becomes F, D becomes G, etc. The last three letters in the alphabet are wrapped around to the beginning: X becomes A, Y becomes B and Z becomes C. Non-letter characters are not modified by the cipher.

Write a program that implements a Caesar cipher. Allow the user to supply the message and the shift amount, and then display the shifted message. Ensure that your program encodes both uppercase and lowercase letters. Your program should also support negative shift values so that it can be used both to encode messages and decode messages.

Problem 7: Square Root

Write a program that implements Newton's method to compute and display the square root of a number, x, entered by the user. The algorithm for Newton's method follows:

Read x from the user
Initialize guess to x/2
While guess is not good enough do
    Update guess to be the average of guess and x/guess

When this algorithm completes, guess contains an approximation of the square root of x. The quality of the approximation depends on how you define "good enough". In the author's solution, guess was considered good enough when the absolute value of the difference between guess * guess and x was less than or equal to 10^{-12}.

Problem 8: Temperature Conversion Table

Write a program that displays a temperature conversion table for degrees Celsius and degrees Fahrenheit. The table should include rows for all temperatures between 0 and 100 degrees Celsius that are multiples of 10 degrees Celsius. Include appropriate headings on your columns. The formula for converting between degrees Celsius and degrees Fahrenheit can be found on the Internet.

Problem 9: Binary to Decimal

Write a program that converts a binary (base 2) number to decimal (base 10). Your program should begin by reading the binary number from the user as a string. Then it should compute the equivalent decimal number by processing each digit in the binary number. Finally, your program should display the equivalent decimal number with an appropriate message.

Problem 10: Decimal to Binary

Write a program that converts a decimal (base 10) number to binary (base 2). Read the decimal number from the user as an integer and then use the division algorithm shown below to perform the conversion. When the algorithm completes, result contains the binary representation of the number. Display the result, along with an appropriate message.

Let result *be an empty string*
Let q *represent the number to convert*
repeat
    Set r *equal to the remainder when* q *is divided by 2*
    Convert r *to a string and add it to the beginning of* result
    Divide q *by 2, discarding any remainder, and store the result back into* q
until q *is 0*