**Task 1: Math Operations and Data Types**

Write a Rust program that:

1. Prompts the user for two numbers, one integer and one floating-point.
2. Perform the following operations:
   * Addition, Subtraction, Multiplication, Division, and Modulus for both types.
3. Print results in a formatted manner (use precision for floats and appropriate padding for integers).
4. Convert between integer and floating-point and demonstrate the result.

**Task 2: Working with Arrays and Tuples**

Create a Rust program that:

1. Initializes an array of 10 integers and a tuple containing an integer, a float, and a character.
2. Perform the following:
   * Replace a few values in the array and print the updated array.
   * Access and print each value from the tuple using both tuple indexing and destructuring.
3. Calculate the sum of all values in the array, find the minimum, maximum, and mean, and display them.

**Task 3: Bitwise and Boolean Operations**

Create a Rust program that:

1. Initializes two integer variables with binary values.
2. Perform bitwise operations (AND, OR, XOR, NOT, Left Shift, Right Shift) on them and print the results in both binary and decimal format.
3. Use boolean variables to perform basic boolean logic operations (AND, OR, XOR, NOT) and print the results.

**Task 4: Loops and Conditionals with Arrays**

Create a Rust program that:

1. Initializes an array of letters (characters).
2. Use a loop to:
   * Print each letter until you encounter a specific letter (e.g., stop at 'l').
3. Implement a conditional expression to print whether the array contains more vowels or consonants.
4. Create a loop that calculates the factorial of a number using both while and for loops, and print the results.

**Task 5: Functions and Closures**

Write a Rust program that:

1. Implements a function calculate that takes two parameters (an integer and a float) and returns the sum, difference, and product as a tuple.
2. Write a recursive function factorial to calculate the factorial of a number and call this function with an input from the user.
3. Write a function apply\_operation that takes two integers and an operator (addition, multiplication, or subtraction) as a string, performs the operation, and returns the result.
4. Print the results from all the function calls in a formatted way.