Task #1: Basic Bitwise Operations

Write a program that takes two integers as input and performs the following bitwise operations. Print the results of each operation on a separate line.

- Bitwise AND (&)
- Bitwise OR (|)
- Bitwise XOR (^)
- Bitwise NOT (~) (Apply NOT to the first integer only)
- Left Shift (<<) (Shift the first integer left by 2 bits)
- Right Shift (>>) (Shift the second integer right by 1 bit)
- **Sample Input:** num1 = 10 (binary 1010), num2 = 6 (binary 0110)
- **Sample Output:**

```
num1 & num2 = 2 (binary 0010)
num1 | num2 = 14 (binary 1110)
num1 ^ num2 = 12 (binary 1100)
~num1 = -11 (binary ... depends on your system's representation of negative
numbers)
num1 << 2 = 40 (binary 101000)
num2 >> 1 = 3 (binary 0011)
```

Task #2: Using Function Pointers for Arithmetic Operations

1. Write functions for addition, subtraction, multiplication, and division. Each function should take two integer arguments and return the result of the operation.
2. Use a function pointer to call these functions dynamically based on user input. Prompt the user to enter the desired operation (+, -, *, /). Then, use the function pointer to call the appropriate function.

- **Sample Input:**

```
Enter first number: 10
Enter second number: 5
Enter operation (+, -, *, /): *
```

- **Sample Output:**

```
Result: 50
```

- **Another Sample Input:**

```
Enter first number: 10
Enter second number: 2
Enter operation (+, -, *, /): /
```

- **Sample Output:**

```
Result: 5
```