Solutions to Logical Problems

1. Sum of First n Numbers

Problem: Print sum of first n numbers.

```
# Function to calculate sum of first n numbers
def sum_n(n):
    return sum(range(1, n + 1))

# Example
n = 5
print("Sum of first", n, "numbers is:", sum_n(n))
```

Output:

```
Sum of first 5 numbers is: 15
```

2. Odd/Even Numbers from 1 to n

Problem: Print list of odd and even numbers from 1 to n.

```
# Function to find odd and even numbers
def odd_even(n):
    even_nums = [i for i in range(1, n + 1) if i % 2 == 0]
    odd_nums = [i for i in range(1, n + 1) if i % 2 != 0]
    return even_nums, odd_nums

# Example
n = 10
evens, odds = odd_even(n)
print("Even numbers:", evens)
print("Odd numbers:", odds)
```

Output:

```
Even numbers: [2, 4, 6, 8, 10]
Odd numbers: [1, 3, 5, 7, 9]
```

3. Pattern with While Loop

Problem: Use while loop to print pattern with sets of 5 numbers and *.

```
# Function to print pattern
def pattern(n):
    i = 1
    while i <= n:
        if (i - 1) % 10 < 5:
            print(i, end=" ")
        else:
            print("*", end=" ")
        i += 1
    print()

# Example
pattern(19)
```

Output:

```
1 2 3 4 5 * * * * * 11 12 13 14 15 * * * *
```

4. Factorial of n

Problem: Print factorial of a number n.

```
# Function to calculate factorial
def factorial(n):
    fact = 1
    for i in range(1, n + 1):
        fact *= i
    return fact
```

```
# Example
n = 4
print("Factorial of", n, "is:", factorial(n))
```

Output:

```
Factorial of 4 is: 24
```

5. Prime Number Check

Problem: Check whether a number is prime or not.

```
# Function to check prime number
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

# Example
n = 6
print("Is", n, "a prime number?:", is_prime(n))
```

Output:

```
Is 6 a prime number?: False
```