These question will help test your understanding of the recursion material presented in class and in the text. These questions are only a study guide. Questions found here may be found on your exam, although perhaps in a different format. Questions NOT found here may also be on your exam.

1. Describe the fundamental structure of any correctly written recursive function.

2. Define **linear recursion**. Give an example of a common mathematical function which exhibits linear recursion.

3. Define **tree recursion**. Give an example of a common mathematical function which exhibits tree recursion.

4. Explain what is meant by a “pending operation”. How are pending operations related to **linear** and **tree** recursion?

5. The well-known Fibonacci sequence is 1, 1, 2, 3, 5, 8, 13, 21, ... and is given by the following mathematical definition

\[
Fib(n) = \begin{cases} 
1, & \text{if } n = 0 \text{ or } n = 1 \\
Fib(n - 1) + Fib(n - 2), & \text{otherwise}
\end{cases}
\]

Write a recursive C++ function that calculated the *nth* Fibonacci number.

6. Write a recursive function that counts the number of 1s in the binary representation of an integer, *N*. (Hint: Think about how many 1s there might be in *N*/2).

7. Write a recursive function named `pattern` whose prototype is

```c++
void Pattern(int little, int big). The function outputs a pattern of integers as shown by the examples below.
```

- `pattern(1, 5)` produces the output 5, 4, 3, 2, 1, 2, 3, 4, 5
- `pattern(17, 20)` produces the output 20, 19, 18, 17, 18, 19, 20
8. What's wrong with the following code that attempts to calculate \( n! \).

```c
int Factorial(int n)
{
    if (n == 0) return 1;
    else return n * (n - 1) * Factorial(n - 2);
}
```

9. What output is produced by the following program?

```c
int mystery(int n)
{
    if (n == 0) return;
    for (int i = 0; i < n; i++)
    {
        cout << n;
        mystery(n - 1);
    }
    cout << endl;
}

int main()
{
    mystery(3);
    return 0;
}
```