CS20 – Section 093 – Fall 2019
Programming Assignment 3 – Due Wednesday September 25 at 11:59PM

Recursion (30 points)

Use the supplied test program to run and debug your code. See Canvas for a copy. Also copied below.

1. Code a recursive function which returns the factorial of a number. The prototype is:

```
int  fact (int)
```

The number to calculate the factorial of is the parameter. The return value is the factorial.

2. Code a recursive function which returns a number in the Fibonocci sequence. The prototype is:

```
int  fib (int)
```

The parameter is the sequence number of the desired Fibonocci number.

3. Code a function which displays the first n prime numbers. The prototype is:

```
void  prime (int)
```

The number of primes to display is passed as the parameter. The function “prime” itself is not recursive. However, it should call a separate recursive helper function which determines if a given number is prime.

4. Code a function named mult which multiplies two numbers and returns the result. The prototype is:

```
int  mult (int, int)
```

The two numbers to multiply are passed as parameters. The return value of the function is the product. Use repetitive addition to perform the multiplication. Assume that both numbers are not negative. Hint: Like for the prime numbers problem, “mult” is not recursive. However, it should call a recursive helper function which performs the multiplication.

5. Code a function which reverses a string. The prototype is:

```
string  reverseString (string)
```

The string passed as a parameter is the one to reverse, and the reversed string is the return value. The original string should remain unchanged. reverseString calls a recursive helper function to perform the reversal.

6. Code a recursive function which counts the number of occurrences of a given character in a string. The prototype is:

```
int  countOccurrences (string, string)
```

The first parameter is the string to search, and the second parameter is a string containing the single character to look for. countOccurrences calls a recursive helper function which performs the count.