Pre-lab Part 1

1.
This function approximate e^x using a for loop.

```
Function Exp(argument a) {
	Sets the variable e to 0
	For variable i =0 is less than 100; i decrements by 1 {
		e plus equals the x^i/factorial of i
	}
	Returns e
end
}
```

2.
This function prints the table for the e^x from the library and the taylor series approximation and their differences.

```
Function Expt(void) {
	Prints x\t\tExp\t\tLibrary\t\tDifference\n
	Prints -\t\t---\t\t-------\t\t----------\n
	For the variable x = 0 to when x is less than or equal to 10, x increments by .1 {
		Prints the x value, calculated Exp value of x, the exp value of x from the math.h
library, and their difference with the absolute value of (Exp(x)-exp(x))
	}
}
\t = tab
\n = new line
```

Pre-lab Part 2

1. The getop() returns the value that is the pointer to the external variable optarg if the option takes a value. It returns -1 if there are no more options to process. It returns ? if there is an unrecognized option.
2. In this case, an enum is much better because it is able to take in more choices like s,c,t,e,a to have multiple options for the program to execute while bool is only have to have two choices because a bool can either be true or false.
3.

```
In the main function (int argc, character **argv) {
	int c is defined
	While the c = getopt of argc, argv, ‘sctea’ does not equal -1 {
		Switch of c {
			Case of ‘s’:
				Prints the sin table
```

```
                    End
            Case of ‘c’:
                    Prints the cos table
                    End
            Case of ‘t’:
                    Prints the tan table
                    end
            Case of ‘e’:
                    Prints the exponential table
                    end
            Case of ‘a’:
                    Prints the sin, cos, tan, exponential table
                    end


        }
    }
    Return 0
    end
}
```

For this assignment, you run the program using ./math.c -’string’ such as s, c, t, e, a. In the main function, it is explained in pre-lab part 2 question 3. It basically takes in the command argument and prints the tables accordingly.

For the sin, cos and exponential values we had to calculate, I used taylor series approximation

For cos, I took the summation of 0=n to 100 of (−1)^n (x^2n /(2n)!)

```
Function Cos(double x) {
    Double c = 0
    For the variable i =0 to 100, i increments {
        C += (−1)^n (x^2n /(2n)!)
    }
    Return c
}
```

For sin, I took the summation of 0=n to 100 of (−1)^n (x^2n+1/ (2n + 1)!)

```
Function Sin(double x) {
    Double s = 0
    For the variable i =0 to 100, i increments {
        s +=(−1)^n (x^2n+1/ (2n + 1)!)
```

```
        }
        Return s
}
```

Where x is the value from -2pi to 2pi for sin and cos with increments of pi/16

For exponential, I took the summation of 0=n to 100 of x^n/n!

```
Function Exp(double x) {
        Double e = 0
        For the variable i =0 to 100, i increments {
                e += x^n/n!
        }
        Return s
}
```

Where x is the value from 0 to 10 for exponential with increments of .1

For the tan values that we had to calculate, I used pade approximation

```
Function Tan(double x) {
        Double t = x(x^8 −990x^6 +135135x^4 −4729725x^2 +34459425)
        Double td = 45(x^8 −308x^6 +21021x^4 −360360x^2 +765765)
        Return t/td
}
```

Where x is the value from (-pi/2)-.001 to (pi/2)+.001 with increments of pi/16

Then I used the values calculated and took the absolute value of the difference against the library values from math.h

Prints out the exponential table

```
Function Expt(void) {
        Prints x\t\tExp\t\tLibrary\t\tDifference\n
        Prints -\t\t---\t\t-------\t\t----------\n
        For variable x = 0 to x <= 10 where x increments by .1 {
                Prints the x value, calculated Exp value of x, the exp value of x from the math.h
library, and their difference with the floating point absolute value of (Exp(x)-exp(x))
        }
}
```

Prints out the Sin table

```
Function Sint(void) {
```

```
    Prints x\t\tSin\t\tLibrary\t\tDifference\n
    Prints -\t\t---\t\t-------\t\t----------\n
    For variable x = -2pi to x <= 2pi where x increments by pi/16 {
        Prints the x value, calculated Sin value of x, the sin value of x from the math.h library, and their difference with the floating point absolute value of (Sin(x)-sin(x))
    }
}

Prints out the Cos table
Function Cost(void) {
    Prints x\t\tCos\t\tLibrary\t\tDifference\n
    Prints -\t\t---\t\t-------\t\t----------\n
    For variable x = -2pi to x <= 2pi where x increments by pi/16 {
        Prints the x value, calculated Cos value of x, the cos value of x from the math.h library, and their difference with the floating point absolute value of (Cos(x)-cos(x))
    }
}

Prints out the Tan table
Function Tant(void) {
    Prints x\t\tTan\t\tLibrary\t\tDifference\n
    Prints -\t\t---\t\t-------\t\t----------\n
    For variable x = (-pi/2)-.001 to x <= (pi/2)+.001 where x increments by pi/16 {
        Prints the x value, calculated Tan value of x, the tan value of x from the math.h library, and their difference with the floating point absolute value of (Tan(x)-tan(x))
    }
}
```