A. Polynomials

Prepare a C program that will calculate the value of the math function:

$$y = 5x^5 + 4x^4 + 3x^3 + 2x^2 + x + 4.$$

Your code should:

1. Set up the program to make use of the C math functions and declare two integer variables for x and y. (Use whatever variable names that you like.)
2. Uses `scanf()` to allow the user (uou) to enter an integer value for x. Include prompts so that the user knows what is to be entered. In using your program, you should keep x between –10 and 10.
3. Calculates the value of y using the math function `pow( )`.
4. Displays the entered value of x and the calculated value of y.

B. Taylor expansion

(Similar to part A.)

Prepare a C program that that will calculate the value of the math function:

$$y = x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1.$$

(Depending on where you are in your calculus classes, you might recognize this as the Taylor expansion for e^x. If not, then don't worry about — just treat this as another random assignment and follow the instructions below.)

Your code should:

1. Set up the program to make use of the C math functions and declares three *double* variables. (Use whatever variable names that you like.)
2. Uses `scanf()` to allow the user (uou) to enter a double value for x. Include prompts so that the user knows what is to be entered. The Taylor expansion above is accurate for values of x near 0, so in using your program, you should enter values of x between –0.5 and 0.5.
3. Calculates the value of y using the power math function and then calculates the exact value of e^x using the exponential function.
4. Displays the entered value of x and the two calculated values, formatted so that the user can easily understand the output.

C. Random numbers and some other math functions

Prepare a C program that implements the following instructions. Your code should:

1. Set up the program to make use of the C math functions *and* the random function. Then declares three *double* variables for x and y. (Use whatever variable names that you like.)

   Note: We will not worry about seeding the random number generator to create a truly random — we will get to that later. For now, use the random function as is.

2. Use the `rand()` function along with modulo division to generate a "random" value between -10 and +10. Print out the value of the random number as soon as it is generated.
3. Calculates square-root of the random number using the `sqrt()` function. Use the `abs()` function to turn any negative values into positive values before taking the square-root. Print out the result.
4. Calculates the base-10 logarithm of the random number (same value as used for part 3). (Note: Taking the logarithm is not defined for negative values, so you will need to take care of that here, as well.) Print out the result.
5. Calculates both the hyperbolic sine and hyperbolic cosine of the random number. (The functions are OK with negative inputs.) Print out the results.

D. Quiz

Finally, there will a short quiz covering some aspects of math functions.

"Reporting"

The three programs (A, B, and C above) can be written prior to lab. You can demo the programs on your own laptop brought to lab or using VirtualBox in the lab room — it's your choice. Your instructor will first ask you to demonstrate one of the programs. A correctly functioning program will earn 15 points. The instructor will then ask you to make one or two modifications to your program. Then you will demo the modified program. If you successfully modify the program, then you will earn an extra 10 points for a total of 25 for the program. You should then print out and submit a written copy of your original program. (You can print out your original version prior to lab, if you are 100% sure that it works properly.)

The process is then repeated for the second and third programs, which are also worth 25 points each.

The quiz is worth 25 points, so that the entire lab is worth 100 points.