C Programming Basics

1. Read in 20 float numbers into an array. (this code is provided to you).
2. For each value in float array, convert as follows:
   * Numbers greater than of equal to 15 should be multiplied by Π squared;. Note that Π is in the file math.h as M\_PI.
   * Numbers not greater than or equal to 15 should converted to their absolute value using the standard C function used for that purpose.
3. Declare an array of 20 integers. Assign each value to the floating point value of the same index (in the floating point array) rounded to the nearest integer.
4. Implement a function (in ics2305-assign1-support.c) to print out the values of the float and integer arrays, one per line, as shown in the example output. Each function should take two values, the array length and the array itself. The functions are defined float\_display\_array (this function should print out the float values to three decimal places, one per line), and integer\_display\_array (this should be print the integer values, one per line). The floating point number should print out 3 digits to the right of the decimal place. Call both functions in the main function using the above defined arrays as parameters.
5. Create two functions that take an array and prints out the number of even values. The first function float\_evens should ignore the part of the number to the right of the decimal point to determine if it is even. The second function integer\_evens should count the number of even numbers as normal. Call both of these functions from the main function inside a single print statement annotating the returned values.
6. Declare another array of 25 integers. Create a function make\_array that receives four parameters:
   * arr - the array itself
   * range - the number of elements in the array
   * exp - a value to use in an exponent
   * md - a modulus value
7. Fill each value of index i as using the following formula: array[i] = exp^i mod md. For example, if exp is 3 and mod is 17, index 4 would be (3^4)%17 = 13. Print each value as you assign it. Hint: you can use the C function pow() to exponentiate.
8. Write a function most\_values to figure out which values occur in the integer array calculated in the preceding step most frequently. The function will receive three parameters:
   * arr - the array itself
   * range - the number of elements in the array
   * maxval - the largest possible value in the array
9. The function will print out the value which occurs the most times in the array. If there are more than one that occur the highest number, print them all. Hint: You can assume that maxval will never exceed 50.
10. Create the graph\_functions function that receives two floating point inputs. This function will graph y=cos(x)\*mult1 where mult1 is the first floating point number passed into the graphing function, and y=sin(x)\*mult2 where mult2 is the second floating point number. You will use text to graph the function as provided by the C printf function. The Y axis should go from 3.0 to -3.0 at -0.1 increments, and the X axis should go from -3.0 to 3.0 at 0.1 increments. Label the graph origin and X=0 and Y=0 coordinates and axes as shown in the sample output (do not overwrite the function values).
11. Add comments to all of your files stating what the code is doing. Fill out the comment function header for each function you are defining in the code. A sample header you can copy for this purpose is provided for the main function in the code.
12. The assignment starter package also includes a sample input (test-input1.txt) and output test-output1.txt which you can use to test your program. The test-input.txt file was input used to produce test-output1.txt. To test your program with these inputs, run the code an pipe in the input file to the program as follows:
    * ./ics2305-assign1 < test-input1.txt
13. Please try to make the output for your program at close to that of the test program output.

Note:

1. You are to complete the ics2305-assign1 program
2. Complete the code in the ics2305-assign1.c and ics2305-assign1-support.c files. Locations where code or declarations need to be added or completed are indicated by ???.
3. Write code in ics2305-assign1-support.c for each of the functions defined in ics2305-assign1-support.h. You are also to complete the function declarations in ics2305-assign1-support.h.