Discussion Think-Pair-Share Activity:

Extend the menu-driven floating-point calculator which three advanced functions.
Part1: square root approximation. We will use a binary search approximation technique for this assignment.

The pseudo-code of the algorithm can be written as follows:
Start with a range of 0 to N
As long as the range is not accurate enough, repeat the following steps:
Compute the middle of the range
Compare the square of the middle value with N
If the middle value is less than the square root
Use middle-to-right as the new range
Otherwise
Use left-to-middle as the new range
Output the middle of the latest range as result

For example, to compute the square root of 10, the program will start with 5, which is in the middle between 0 and 10. Since $5 * 5 = 25$ is larger than 10, the program will try the middle number 2.5 of left bound (0 to 5). Thus, the program compares $2.5 * 2.5$ with 10. Because the result 6.25 is smaller than 10, it will pick 3.75 (the middle number of 2.5 and 5) as the next guess. By picking the middle number every time and comparing its square with the original number, the program gets closer to the actual square root.

To design this program, let us take a look at the following questions before we start programing.

1. What is the input and output of this program for this assignment?

   Input: the current result value
   Output: the approximate square root, the approximate results for each step, etc.

2. You need to implement this approximation as a function:
   /*Get the approximate square root return the value*/
   double ApproximateSquareRoot(double x);
   What does this function return? What is the argument for this function when you call it? When will you call this function? How will you call this function?

   The approximate square root result.
   The argument is currValue.
   Call this function for option 6.
   currValue = ApproximateSquareRoot(currValue);

3. How many variables will you need? What are they? What are the types? What are their scopes?

   Number of interations, integer
   Approximate square root for each step, one variable, double
   Left and right boundaries, double
   Etc.

4. We will use the binary search technique to get the approximate square root. Can you simulate the program by finding the square root of 64? How many iterations will you have? What are the left and right boundaries for each search step?

   Discussed in the discussion session.

5. Which control flow will you use to implement this function?

loops

6. How to check whether the value of a floating point variable is 0 or not in C program? (HINT: floating point variables cannot represent 0 precisely. You need approximate comparison. More details are in the assignment handout)

   Check whether the searching range is small enough, i.e. R-L<0.00001 (R>L)