1. Write a C program that prompts the user inputting two real numbers as length for the two legs of a right triangle and makes use of the ***pow( )*** and ***sqrt( )*** functions and the Pythagorean theorem to compute the length of the hypotenuse. This program should consider any possible error cases and abnormal input, and output calculation results as well as error messages with reason. (10 points)

2. Implement some mathematical functions in C language without calling any functions in the standard C library. (40 points, 10 points for each)
   (1). Implement a recursive version ***pow_recursive(double x, int y)*** to compute the power value of x to the exponent of positive integer y. How is the time complexity of your implementation?
   (2). Implement the function ***exp(double x)*** by referring to the equation $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} \cdots$. You should calculate at least 100 terms.
   (3). Implement the natural logarithm function ***log(double x)*** by referring to the equation $\log(x) = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \frac{(x-1)^5}{5} \cdots$. You should calculate at least 100 terms.
   (4). Implement the power function ***pow(double x, double y)*** by calling all of your previous functions. First, you should consider a possibly negative exponent, and decompose the exponent into an integer part and a rational part. For example, $2^{-3.5} = (2^3 \cdot 2^{0.5})^{-1} = \frac{1}{(2 \cdot 2 \cdot 2) \cdot 2^{0.5}}$. Then, for the rational part, you shall refer to the equation $x^y = e^{y \log x}$. For example, $2^{0.5} = e^{0.5 * \log 2}$. You might get no points if you don't follow the above instructions.

3. There is an integer array with the size of 100. (1). Write an algorithm to remove duplicate values. (2). Use either C or Java to implement the algorithm. (20 points, 10 points for each)

4. Suppose that you have a business with several offices. You want to connect all offices with each other via leased lines, which have different amounts of money for different pairs of cities. (1). Which data structure will you use in your network design to connect all your offices with a minimum total cost? (2). Write an example with a short description (around 50 English words) to illustrate how you use this data structure in your network design with a minimum total cost. (3). Which data structure will you use in your network design to connect all your offices with a maximum total cost? (30 points, 10 points for each)