Develop the following programs. Any global variable, static variable, and array CANNOT be used (unless specified). You also need to finish each step as you did in the first project.

Sample: Calculate the result 1+2+3+4+5+6+…+99 (no need to use print inside function).

1. What's the biggest job overall (explain it in your own words)?
   1+2+3+4+5+6+…+99

3. What do you expect the computer do for the first division
   job division: 1+2+3+4+5+6+…+99+98+99 => 1+2+3+4+5+6+…+98
   \( f(99) \) => \( f(98) \)
   results collection: \( f(98)+99 \)
   result reporting: return

(2) What do you expect the computer do for the second division
   job division: 1+2+3+4+5+6+…+97+98 => 1+2+3+4+5+6+…+97
   \( f(98) \) => \( f(97) \)
   results collection: \( f(97)+98 \)
   result reporting: return

(2) What do you expect the computer do generally at each division
   (job description, job division, results collection, and result reporting, in general format)?
   --- Pay attention on how to reuse the existing result from the above step.
   job division: 1 to \( x \) => 1 to \( x-1 \)
   \( f(x) \) => \( f(x-1) \)
   results collection: \( f(x-1)+x \)
   result reporting: return

1. What's the terminating condition?
   If \( (x<=1) \)

1. What's the program (including function declaration, and function call)
   int \( f(x) \) {
     if \( (x <=1) \) return 1;
     else return \( f(x-1)+x \);
   }

1. Calculate the result 1+2+4+8+16+32+…+4096. No need to print inside function.
2. Calculate the result 1/i+2/(i-1)+3/(i-2)+…+(i-1)/2+i/1. No need to print inside function.
3. The world population reached 6 billion people in 1999 and grows at the rate of 1.4 percent each year. Write a recursive function \( f \) to return/determine when the population will exceed 10 Billion.

4. (Array is needed!) Write a function to print all of the one-element sequences of a char [] array, then all the two-element sequences, and so on. The characters in each sequence appear in the same order as they are in the array. For example, for the array char [] letters {'A', 'C', 'E', 'G'}; the one-element sequences are “A”, “C”, “E”, and “G”; the two-element sequences are “AC”, “AE”, “AG”, “CE”, “CG”, and “EG”; the three-element sequences are “ACE”, “ACG”, “AEG”, “CEG”; the four-element sequence is “ACEG”

5. Develop a recursive function \( f \) to display the sequence
   1 2 3 4 5 … i (i-1) (i-2) … 2 1 in a text field when \( f(1,i) \) is called.