1. **Give a) an example of a valid instance of the summation problem that is not a valid instance of the minimum problem, and b) vice-versa.**
   a) input : empty list ([ ])
      output : 0 for summation, error due to invalid instance for minimum.
   explanation: In summation, any empty list of numbers is valid, but for minimum, it requires a non-empty list instead, and so an error would occur as there are no numbers to compare each other with.

   b) input: a list of chars (['a',''b','c','d'])
      output: 'a' for minimum, error due to invalid instance for summation only accepts numbers
   explanation: In a minimum problem, a list of strings ['a', 'b', 'c','d'] is valid, but for summation, it requires a list of numbers instead, and so an error would occur since the input can't be summed up.

2. **Each of the following snippets of pseudocode fails to live up to all of the clarity, correctness, and termination requirements of algorithms. In each case, describe the failing, and then rewrite the pseudocode as a proper algorithm. Hint: consult the checklist in section 2.3.**
   a) **for list:**
      total = total + i
      describe the failing: The function is missing inputs and outputs, it doesn't return anything, it uses undefined variables (i, total), and the function definition and name are missing

      ```
      def sumItemsInList(list):
          total = 0
          for i in list
            total = total + i
            return total
      ```

   b) **def long_division (numerator, denominator):**
      quotient = numerator / denominator
      remainder = numerator % denominator
      describe the failing: The function doesn't return anything, it's using undefined variables (quotient, remainder) and the bases cases are not covered.

      ```
      def long_division (numerator, denominator):
          quotient = 0
          remainder = 0
           if(numerator  == 0) return (0,0)
           if(denominator == 0) return none
          quotient = numerator / denominator
          remainder = numerator % denominator
           return (quotient,remainder)
      ```

3. **Write a problem definition for each of the following problems.**
   a) **Determining whether every element in a sequence is identical**
      all elements identical in sequence problem
      intput: a non-empty list L of n comparable objects, n>=2
      outputs: if two or more elements are not identical or n<2 return false, else return true

   b) **Determining whether two strings are identical**
      identical string problem
      input: two strings str1and str2
      output: if str1 is identical to str2 return true, else return false