1. (12 points) Answer the following questions about languages and grammars.

   a) Is the C programming language a context free language? Explain.

   b) Can the same language have an ambiguous and an unambiguous grammar at the same time? Explain.

   c) Write a grammar for strings with matching quotes where $\Sigma = \{a, b, \}'$. E.g. 'aba', aba"bb, aa’a’a’a’.

   d) Write a grammar for strings with matching parentheses $\Sigma = \{a, b, (, )\}$. E.g. (aba), aba(bb, aa(a(a)a).

   e) Write a regular expression for the language expressed in c), if possible. If not possible, explain in your own words why not.

   f) Write a regular expression for the language expressed in d), if possible. If not possible, explain in your own words why not.

2. (10 points) Given the following grammar, construct the first and follow sets for each non-terminal symbol.

   $A \rightarrow BAe \mid FE$
   $B \rightarrow bEF \mid g$
   $E \rightarrow e \mid \varepsilon$
   $F \rightarrow f \mid EH$
   $H \rightarrow h$
3. (16 points) For each of the below grammars, do the following:
   1) Determine whether the grammar is LL(k). If not, explain using First and Follow sets.
      (LL(k) is a predictive parser with arbitrary lookahead.)
   2) Determine whether the grammar is unambiguous. If not, give an example input with two
      parse trees.
      a) $S \rightarrow [S \mid A$
         $A \rightarrow [A] \mid \varepsilon$
      b) $S \rightarrow ABc$
         $A \rightarrow a \mid \varepsilon$
         $B \rightarrow b \mid \varepsilon$
      c) $S \rightarrow ABBA$
         $A \rightarrow a \mid \varepsilon$
         $B \rightarrow b \mid \varepsilon$
      d) $S \rightarrow aA\varepsilon|bA\varepsilon$
         $A \rightarrow b \mid \varepsilon$

4. (10 points) Given the following grammar, answer the below questions:
   $E \rightarrow E + E \mid id$
   a) The above grammar is not LL(1). Calculate First and Follow sets to identify the input
      symbol with the conflict.
   b) Write a new grammar after performing left-recursion removal.
   c) The grammar in b) is still not LL(1). Calculate First and Follow sets to identify the
      input symbol with the conflict.
   d) Modify the original grammar such that the + operator is left associative, and then
      perform left-recursion removal. Write the new grammar.
   e) Show the grammar in d) is now LL(1) by drawing the LL(1) parse table below.