Assignment : 3

1. Write an algorithm for eliminating left recursion.
2. Which of the following grammer are ambiguous?Justify your answer.

   a) S -> a | Sa | bSS | SSb | SbS

   b) S-> a| S+S | SS | S* | (S)

   c) S-> S(S)S | ^

   d) S-> aS | aSbS | ^

   e) S-> SS+ | SS- | a

3. Eliminate left recursion and perform left factoring on given grammer.

   a) S -> A
      B -> bBc | f
      A -> Ad | Ac | aB | ac

   b) E -> Ma | Sb
      M -> ES | ah
      S -> ShE | ^

   c) A -> ad | a | ab | abc | b.

4. Construct predictive parsing table for following:

   S -> A

   A -> aB | Ad

   B -> bBC | f

C -> g

5. Construct a recursive decent parser with backtracking for the following grammer:

   S -> aSbS | bSaS | ^

   Parse the string with backtracking: aabb$

6. Find first and follow for given grammer.

   a) A -> (A)A | ^

   b) S -> ACB | cbB | Ba

      A -> da | BC

      B -> g | ^

      C -> h | ^

7. Find Whether the given grammer is LL(1) or not:

   a) S -> 1AB | ^

      A -> 1AC | 0C

      B -> 0S

      C -> 1

   b) A -> BCx | y

      B -> yA | ^

      C -> Ay | x