1. [60] Consider the following grammar G
   S → Yz
   Y → A|B | 2
   A → 2
   B → A
   where A, B, and Z are nonterminals, 1, 2 and z are terminals
   (a) What is the language L(G) generated by G
   (b) Is G LL(1)? If not, why? If yes, show the parsing table.
   (c) Is G SLR(1)? If not, why? If yes, show the parsing table.
   (d) Is G LR(1)? If not, why? If yes, show the parsing table.
   (e) Is G LALR(1)? If not, why? If yes, show the parsing table.
   (f) Is G operator-precedence? If not, why? If yes, show the parsing table.

2. [20] Consider the following grammar G
   S → (L) | a
   L → L , S | S
   (a) Rewrite G to G' to eliminate left recursion
   (b) Write down the FIRST and FOLLOW sets for all nonterminals of G'
   (c) Show the predictive parsing table of G'
   (d) Show the process of parsing the string "(a, (a, a)) S" by the predictive parser

3. [10] Why there will be no shift-reduce conflicts introduced by merging sets of LR(1) items with the same core?

4. [20] Consider the following Pascal program. What is the output
   (a) under static scope?
   (b) under dynamic scope?
   (c) How to implement dynamic scope?
   (d) Please show the activation records in the control stack when n is printed under dynamic scope.

```
program scope
  var n: integer;
  procedure show;
    begin write(n) end;
  procedure small;
    var n: integer;
    begin n:= 3; show end;
  begin
    n := 5;
    show; small
  end.
```