Name : Roy Ivey
Course: CS 4280 Program Translation
Assignment: Homework 1

1.
Part 1:
Grammer with Left Recursion Removed

```
goal  → expr

expr  → term expr' | term
expr' → + term | - term

term  → term term' | fact
term' → * fact | / fact | % fact

fact  → IntConst | ref

ref   → Ident( expr ) | Ident
```

Part 2:
Recursive Descent Functions

```
void goal()
{
   expr();
}

void expr()
{
   if (next_token == "-" || next_token == "+") { accept(next_token);  }
   term();
}

void term()
{
   if ( next_token == "*" || next_token = "/" || next_token = "%")
      {accept(next_token);}
   fact();
}

void fact()
{
   if     (next_token == "IntConst") { accept(next_token)         ; }
   elseif (next_token == ")"       ) { accept(next_token)         ; }
   elseif (next_token == "Ident"   ) { ref()                      ; }
   else                              { error("Syntactical error."); }
}

void ref()
{
   accept(next_token);
   if ( next_token == "(" )                       { accept(next_token); expr(); }
   elseif(next_token == ")" or next_token == EOF) { accept(next_token);}
   else                                           { error("Syntactical error"); }
}
```

2. LR0 Full item sets

```
1. goal      → ^decl |-   [2]
   decl      → ^constdecl [3]
   decl      → ^vardecl   [4]
   constdecl → ^const id  :=   expn [5]
   constdecl → ^const id  :    type [5]
   vardecl   → ^var idlist := expn  [6]
   vardecl   → ^var idlist : type   [6]

2. goal      → decl ^ |- [7]

3. decl      → constdecl^

4. decl      → vardecl^

5. constdecl → const ^id := expn [8]
   constdecl → const ^id :  type [8]

6. vardecl   → var ^idlist := expn [9]
   vardecl   → var ^idlist : type  [9]

7. goal      → decl |-^

8. constdecl →  const  id ^:=  expn [11]
   constdecl →  const  id ^:   type [12]

9. idlist    → ^id, idlist [17]
   idlist    → ^id         [8]

10. vardecl   → var idlist ^:= expn [13]
    vardecl   → var idlist ^:  type [14]

11. constdecl →  const  id := ^expn [15]

12. constdecl →  const  id : ^type  [16]

13. vardecl   → var idlist := ^expn [15]

14. vardecl   → var idlist :  ^type [16]

15. expn      → ^number             [18]

16. type      → ^int                [2]
    type      → ^real               [2]

17. idlist    → id, ^idlist [9]
```