Compiler Top-Down Parse Exercise

Summer - 2017

Given this Top-Down parse grammar:

1. <stmt-list> ::= <stmt> { ; <stmt> }
2. <stmt> ::= <assign> | <read> | <write> | <for>
3. <assign> ::= **id** := <exp>
4. <exp> ::= <term> { + <term> | - <term> }
5. <term> ::= <factor> { * <factor> | DIV <factor> }
6. <factor> ::= **id** | **int** | (<exp>)
7. <read> ::= READ (<id-list>)
8. <id-list> ::= **id** { ; **id** }
9. <write> ::= WRITE (<id-list>)
10. <for> ::= FOR <index-exp> DO <body>
11. <index-exp> ::= **id** := <exp> TO <exp>
12. <body> ::= <stmt> | BEGIN <stmt-list> END

Draw the Top-Down parse tree for this <stmt>

```
FOR I := 1 TO 100 DO
        BEGIN
            READ ( value );
            sum   := sum + value;
            sumSQ := sumSQ + value * value
        END
```