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> }

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

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

## FOR I := 1 TO 100 DO

**BEGIN**

**END**

**READ ( value );**

**sum := sum + value;**

**sumSQ := sumSQ + value \* value**