ECE 468 & 573
Problem Set 3: Symbol tables, ASTs and semantic actions

1. Why do we track the number of dimensions and sizes of arrays in symbol tables?

2. What differentiates an abstract syntax tree from a parse tree?

3. Name one advantage to generating ASTs before producing code, rather than producing code directly.

4. Show what the abstract syntax tree would look like for the following expression:

$$w := x + y * (z + 3)$$

5. Give three address code would be generated for the above tree. Use the following instructions: **LD A, T** loads from variable A into temporary T. **OP T1, T2, T3** performs `T3 = T1 OP T2`. **ST T, A** stores from variable A into temporary T.

6. Show the code generation information (any code, what temporary stores the result, and whether it's an l-value or an r-value) for each node in the AST above.