

The Tiny Language CFG

program \rightarrow stmt-sequence

stmt-sequence \rightarrow statement { ; statement }

statement \rightarrow if-stmt | repeat-stmt | assign-stmt | read-stmt | write-stmt

if-stmt \rightarrow **if** exp **then** stmt-sequence [**else** stmt-sequence] **end**

repeat-stmt \rightarrow **repeat** stmt-sequence **until** exp

assign-stmt \rightarrow **identifier** := exp

read-stmt \rightarrow **read identifier**

write-stmt \rightarrow **write** exp

exp \rightarrow simple-exp [comparison-op simple-exp]

comparison-op \rightarrow < | =

simple-exp \rightarrow term { addop term }

addop \rightarrow + | -

term \rightarrow factor { mulop factor }

mulop \rightarrow *

factor \rightarrow (exp) | **number** | **identifier**

Task Statement: Design and implement a top down parser for the above language that returns the syntax tree of a given program.

Note: Terminals (tokens) are written in bold. This is the minimum specs of the language syntax; you are encouraged to add more features for a bonus in return.