- A context-free Grammar for TINY:

program→ stmt-sequence

stmt-sequence → stmt-sequence ; statement | statement

statement→ if- stmt | repeat-stmt | assign-stmt | read-stmt | write-stmt

if -stmt → **if** exp **then** stmt-sequence **end**

|**if** exp **then** stmt-sequence **else** stmt-sequence **end**

repeat-stmt → **repeat** stmt-sequence **until** exp

assign-stmt → **identifier :=** exp

read-stmt → **read identifier**

write-stmt → **write** exp

exp → simple-exp comparison-op simple-exp | simple-exp

comparison-op → **<** | **=**

simple-exp → simple-exp addop term | term

addop → **+** | **-**

term → term mulop factor | factor

mulop → ***** | **/**

factor → **(**exp**)** | **number** | **identifier**

- BNF to EBNF:

program→ stmt-sequence

stmt-sequence → statement { ; statement }

statement→ if- stmt | repeat-stmt | assign-stmt | read-stmt | write-stmt

if -stmt → **if** exp **then** stmt-sequence [**else** stmt-sequence] **end**

repeat-stmt → **repeat** stmt-sequence **until** exp

assign-stmt → **identifier :=** exp

read-stmt → **read identifier**

write-stmt → **write** exp

exp → simple-exp [comparison-op simple-exp]

comparison-op → **<** | **=**

simple-exp → term { addop term }

addop → **+** | **-**

term → factor { mulop factor }

mulop → ***** | **/**

factor → **(**exp**)** | **number** | **identifier**