Modified Tiny-C BNF - Original

```
<program> ::= <statement_list>
<statement_list> ::= <statement> SC <statement_list> | ∈
<statement> ::= <if_statement> |
                WHILE <paren_expr> <statement> |
                DO <statement> WHILE <paren_expr> |
                <expr> |
                SC
<if_statement> ::= IF <paren_expr> <statement>
<else_clause> ::= ELSE <statement> | ∈
<paren_expr> ::= LP <expr> RP
<expr> ::= <test> | id ASGN <expr>
<test> ::= <sum> <test_opt>
<test_opt> ::= COMPARE <sum> | ∈
<sum> ::= <term> <sum_opt>
<sum_opt> ::= ADD <term> <sum_opt> | SUB <term>
<sum_opt> | ∈
<term> ::= id | num | <paren_expr>
```

Modified Tiny-C BNF - Removing ELSE Statements

```
<program> ::= <statement_list>
<statement_list> ::= <statement> SC <statement_list> | ∈
<statement> ::= IF <paren_expr> <statement> |
                WHILE <paren_expr> <statement> |
                DO <statement> WHILE <paren_expr> |
                <expr> |
                SC
<paren_expr> ::= LP <expr> RP
<expr> ::= <test> | id ASGN <expr>
<test> ::= <sum> <test_opt>
<test_opt> ::= COMPARE <sum> | ∈
<sum> ::= <term> <sum_opt>
<sum_opt> ::= ADD <term> <sum_opt> | SUB <term>
<sum_opt> | ∈
<term> ::= id | num | <paren_expr>
```

Modified Tiny-C BNF - Removing ELSE Statements

```
<program> ::= <statement_list>
<statement_list> ::= <statement> SC <statement_list> | ∈
<statement> ::= IF <paren_expr> <statement> |
                WHILE <paren_expr> <statement> |
                DO <statement> WHILE <paren_expr> |
                <expr> |
                SC
<paren_expr> ::= LP <expr> RP
<expr> ::= <test> | id ASGN <expr>
<test> ::= <sum> <test_opt>
<test_opt> ::= COMPARE <sum> | ∈
<sum> ::= <term> <sum_opt>
<sum_opt> ::= ADD <term> <sum_opt> | SUB <term>
<sum_opt> | ∈
<term> ::= id | num | <paren_expr>
```

Modified Tiny-C BNF - Moving ASGN production to the top

```
<program> ::= <statement_list>
<statement_list> ::= <statement> SC <statement_list> | ∈
<statement> ::= IF <paren_expr> <statement> |
                WHILE <paren_expr> <statement> |
                DO <statement> WHILE <paren_expr> |
                id ASGN <expr> |
                SC
<paren_expr> ::= LP <expr> RP
<expr> ::= <test>
<test> ::= <sum> <test_opt>
<test_opt> ::= COMPARE <sum> | ∈
<sum> ::= <term> <sum_opt>
<sum_opt> ::= ADD <term> <sum_opt> | SUB <term>
<sum_opt> | ∈
<term> ::= id | num | <paren_expr>
```

Parsing table

<program> ::= <statement_list> **1**

<statement_list> ::= <statement> SC <statement_list> | **2**

     ∈ **3**

<statement> ::= IF <paren_expr> <statement> | **4**

    WHILE <paren_expr> <statement> | **5**

    DO <statement> WHILE <paren_expr> | **6**

    id ASGN <expr> | **7**

    SC **8**

<paren_expr> ::= LP <expr> RP **9**

<expr> ::= <test> **10**

<test> ::= <sum> <test_opt> **11**

<test_opt> ::= COMPARE <sum> **12**

   | ∈ **13**

<sum> ::= <term> <sum_opt> **14**

<sum_opt> ::= ADD <term> <sum_opt> | **15**

   SUB <term> <sum_opt> | **16**

   ∈ **17**

<term> ::= id | **18**

   num | **19**

   <paren_expr> **20**

	num	id	SUB	ADD	COMPARE	RP	LP	SC	ASGN	WHILE	DO	IF	$
<program>		1						1		1	1	1	1
<statement_list>		2						2		2	2	2	3
<statement>		7						8		5	6	4	
<paren_exp>							9						
<expr>	10	10					10						
<test>	11	11					11						
<test_opt>					12	13		13		13			
<sum>	14	14					14						
<sum_opt>			16	15	17	17		17		17			
<term>	19	18					20						