Example Grammar From Text

LL(1) version:

program ⟶ *stmt_list* `$$`
stmt_list ⟶ *stmt stmt_list* | ϵ
stmt ⟶ `id :=` *expr* | `read id` | `write` *expr*
expr ⟶ *term term_tail*
term_tail ⟶ *add_op term term_tail* | ϵ
term ⟶ *factor factor_tail*
factor_tail ⟶ *mult_op factor factor_tail* | ϵ
factor ⟶ `(` *expr* `)` | `id` | `number`
add_op ⟶ `+` | `-`
mult_op ⟶ `*` | `/`

Additional Pieces (for either):

stmt ⟶ `if` *condition* `then` *stmt_list* `fi`
⟶ `while` *condition* `do` *stmt_list* `od`
condition ⟶ *expr relation expr*
relation ⟶ `<` | `>` | `<=` | `>=` | `=` | `!=`

LR(1) version

program ⟶ *stmt_list* `$$`
stmt_list ⟶ *stmt_list stmt*
stmt_list ⟶ *stmt*
stmt ⟶ `id :=` *expr*
stmt ⟶ `read id`
stmt ⟶ `write` *expr*
expr ⟶ *term*
expr ⟶ *expr add_op term*
term ⟶ *factor*
term ⟶ *term mult_op factor*
factor ⟶ `(` *expr* `)`
factor ⟶ `id`
factor ⟶ `number`
add_op ⟶ `+` | `-`
mult_op ⟶ `*` | `/`

Sample Valid Input:

```
abs := n
if n < 0 then abs := 0 - abs fi

sum := 0
read count
while count > 0 do
    read n
    sum := sum + n
    count := count - 1
od
write sum
```