

```

#!/usr/bin/env python3
# FILE: compiler.py
# FINAL PROJECT
# Professor Ahmadnia
# Group: Kevin Vuong, Anika Corpus, Christopher Grant
# Description: This program reads source code from a file, outputs it w/o comments and
properly formatted spaces.
#           It then tokenizes and then parses the source code and checks if it has
correct grammar.

from text_clean import *

from tokenizer import *

from LL_parser import *

from code_generator import *

def main():
    # First, remove all the comments
    with open("finalv1.txt") as source_file:
        new_file = open('finalv2.txt', mode='w+', encoding='utf-8')
        comment_removal(source_file, new_file)
        new_file.close()

    # Clean the spaces
    content = clean_text('finalv2.txt')

    token_list = tokenizer(content)

    if token_list == -1:
        exit(1)

    terminal_list = ['P', 'Q', 'R', 'S', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
'PROGRAM', 'BEGIN', 'END.',
                    'INTEGER', 'PRINT', '+', '-', '/', '*', '(', ')', ',', ';', '=', ':',
'$']

    predict_table = {
        'W': {'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'aaa', '1': 'aaa', '2':
'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa',
'PROGRAM': 'PROGRAM B ; D BEGIN I END.', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', '
PRINT': 'aaa', '+': 'aaa', '-': 'aaa', '/': 'aaa', '*': 'aaa', '(': 'aaa', ')': 'aaa', ',':
'aaa', ';': 'aaa', '=': 'aaa', ':': 'aaa', '$': 'aaa'},
        'B': {'P': 'UC', 'Q': 'UC', 'R': 'UC', 'S': 'UC', '0': 'aaa', '1': 'aaa', '2': 'aaa
', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa', '
PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+': 'aaa
', '-': 'aaa', '/': 'aaa', '*': 'aaa', '(': 'aaa', ')': 'aaa', ',': 'aaa', ';': 'aaa', '=':
'aaa', ':': 'aaa', '$': 'aaa'},
        'C': {'P': 'UC', 'Q': 'UC', 'R': 'UC', 'S': 'UC', '0': 'OC', '1': 'OC', '2': 'OC',
'3': 'OC', '4': 'OC', '5': 'OC', '6': 'OC', '7': 'OC', '8': 'OC', '9': 'OC', 'PROGRAM': '
aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+': 'lambda', '-':
'lambda', '/': 'lambda', '*': 'lambda', '(': 'aaa', ')': 'lambda', ',': 'lambda', ';': 'aaa
', '=': 'lambda', ':': 'aaa', '$': 'aaa'},
        'D': {'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'aaa', '1': 'aaa', '2':
'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa',
'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'H : G ;', 'PRINT': 'aaa', '+':
'aaa', '-': 'aaa', '/': 'aaa', '*': 'aaa', '(': 'aaa', ')': 'aaa', ',': 'aaa', ';': 'aaa', '=':
'aaa', ':': 'aaa', '$': 'aaa'},
        'G': {'P': 'BZ', 'Q': 'BZ', 'R': 'BZ', 'S': 'BZ', '0': 'aaa', '1': 'aaa', '2': 'aaa
', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa', '
PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+': 'aaa
', '-': 'aaa', '/': 'aaa', '*': 'aaa', '(': 'aaa', ')': 'aaa', ',': 'aaa', ';': 'aaa', '=':
'aaa', ':': 'aaa', '$': 'aaa'},
        'Z': {'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'aaa', '1': 'aaa', '2':
'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa',
'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+': '
aaa', '-': 'aaa', '/': 'aaa', '*': 'aaa', '(': 'aaa', ')': 'aaa', ',': ' ', G', ';': 'aaa',
':': 'lambda', '=': 'aaa', ':': 'aaa', '$': 'aaa'},

```

```
H': { 'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'aaa', '1': 'aaa', '2':  
'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa',  
'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'INTEGER', 'PRINT': 'aaa', '+':  
'aaa', '-': 'aaa', '/': 'aaa', '*': 'aaa', '(' : 'aaa', ')': 'aaa', ',': 'aaa', '.': 'aaa',  
';': 'aaa', '=': 'aaa', ':': 'aaa', '$': 'aaa'},  
I': { 'P': 'JV', 'Q': 'JV', 'R': 'JV', 'S': 'JV', '0': 'aaa', '1': 'aaa', '2': 'aaa'  
, '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa', '  
PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'JV', '+' : 'aaa'  
, '-' : 'aaa', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'aaa', ',' : 'aaa', '.' : 'aaa', ';' :  
'aaa', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'},  
V': { 'P': 'I', 'Q': 'I', 'R': 'I', 'S': 'I', '0': 'aaa', '1': 'aaa', '2': 'aaa', '  
3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa', 'PROGRAM  
' : 'aaa', 'BEGIN': 'aaa', 'END.': 'lambda', 'INTEGER': 'aaa', 'PRINT': 'I', '+' : 'aaa', '-' :  
' : 'aaa', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'aaa', ',' : 'aaa', '.' : 'aaa', ';' : 'aaa'  
, '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},  
J': { 'P': 'A', 'Q': 'A', 'R': 'A', 'S': 'A', '0': 'aaa', '1': 'aaa', '2': 'aaa', '  
3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa', 'PROGRAM  
' : 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'K', '+' : 'aaa', '-' :  
'aaa', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'aaa', ',' : 'aaa', '.' : 'aaa', ';' : 'aaa',  
'=' : 'aaa', ':' : 'aaa', '$' : 'aaa'},  
K': { 'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'aaa', '1': 'aaa', '2':  
'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa',  
'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'PRINT ( B ) ;'  
, '+' : 'aaa', '-' : 'aaa', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'aaa', ',' : 'aaa', '.' :  
'aaa', ';' : 'aaa', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},  
A': { 'P': 'B = E ;', 'Q': 'B = E ;', 'R': 'B = E ;', 'S': 'B = E ;', '0': 'aaa', '  
1': 'aaa', '2': 'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': '  
aaa', '9': 'aaa', 'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT  
' : 'aaa', '+' : 'aaa', '-' : 'aaa', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'aaa', ',' : 'aaa'  
, '.' : 'aaa', ';' : 'aaa', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},  
E': { 'P': 'TX', 'Q': 'TX', 'R': 'TX', 'S': 'TX', '0': 'TX', '1': 'TX', '2': 'TX',  
'3': 'TX', '4': 'TX', '5': 'TX', '6': 'TX', '7': 'TX', '8': 'TX', '9': 'TX', 'PROGRAM': '  
aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+' : 'TX', '-' : 'TX'  
, '/' : 'aaa', '*' : 'aaa', '(' : 'TX', ')' : 'aaa', ',' : 'aaa', '.' : 'aaa', ';' : 'aaa', '=' :  
aaa', ':' : 'aaa', '$' : 'aaa'}},  
X': { 'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'aaa', '1': 'aaa', '2':  
'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa',  
'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+' : '+  
TX', '-' : '-TX', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'lambda', ',' : 'aaa', '.' : 'aaa',  
';': 'lambda', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},  
T': { 'P': 'FY', 'Q': 'FY', 'R': 'FY', 'S': 'FY', '0': 'FY', '1': 'FY', '2': 'FY',  
'3': 'FY', '4': 'FY', '5': 'FY', '6': 'FY', '7': 'FY', '8': 'FY', '9': 'FY', 'PROGRAM': '  
aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+' : 'FY', '-' : 'FY'  
, '/' : 'aaa', '*' : 'aaa', '(' : 'FY', ')' : 'aaa', ',' : 'aaa', '.' : 'aaa', ';' : 'aaa', '=' : '  
aaa', ':' : 'aaa', '$' : 'aaa'}},  
Y': { 'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'aaa', '1': 'aaa', '2':  
'aaa', '3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa',  
'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+' : '  
lambda', '-' : 'lambda', '/' : '/FY', '*' : '*FY', '(' : 'aaa', ')' : 'lambda', ',' : 'aaa', '.' :  
'aaa', ';' : 'lambda', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},  
F': { 'P': 'B', 'Q': 'B', 'R': 'B', 'S': 'B', '0': 'N', '1': 'N', '2': 'N', '3': 'N'  
, '4': 'N', '5': 'N', '6': 'N', '7': 'N', '8': 'N', '9': 'N', 'PROGRAM': 'aaa', 'BEGIN': '  
aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+' : 'N', '-' : 'N', '/' : 'aaa', '*' :  
'aaa', '(' : '( E )', ')' : 'aaa', ',' : 'aaa', '.' : 'aaa', ';' : 'aaa', '=' : 'aaa', ':' : 'aaa'  
, '$' : 'aaa'}},  
N': { 'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'LOM', '1': 'LOM', '2':  
'LOM', '3': 'LOM', '4': 'LOM', '5': 'LOM', '6': 'LOM', '7': 'LOM', '8': 'LOM', '9': 'LOM',  
'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+' : '  
LOM', '-' : 'LOM', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'aaa', ',' : 'aaa', '.' : 'aaa',  
';': 'aaa', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},  
L': { 'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'lambda', '1': 'lambda',  
'2': 'lambda', '3': 'lambda', '4': 'lambda', '5': 'lambda', '6': 'lambda', '7': 'lambda',  
'8': 'lambda', '9': 'lambda', 'PROGRAM': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': '  
aaa', 'PRINT': 'aaa', '+' : '+', '-' : '-', '/' : 'aaa', '*' : 'aaa', '(' : 'aaa', ')' : 'aaa',  
, ',' : 'aaa', '.' : 'aaa', ';' : 'aaa', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},  
M': { 'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': 'OM', '1': 'OM', '2': '  
OM', '3': 'OM', '4': 'OM', '5': 'OM', '6': 'OM', '7': 'OM', '8': 'OM', '9': 'OM', 'PROGRAM'  
' : 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+' : 'lambda',  
-' : 'lambda', '/' : 'lambda', '*' : 'lambda', '(' : 'aaa', ')' : 'lambda', ',' : 'aaa', '.' : '  
aaa', ';': 'lambda', '=' : 'aaa', ':' : 'aaa', '$' : 'aaa'}},
```

```

        'O': {'P': 'aaa', 'Q': 'aaa', 'R': 'aaa', 'S': 'aaa', '0': '0', '1': '1', '2': '2',
        '3': '3', '4': '4', '5': '5', '6': '6', '7': '7', '8': '8', '9': '9', 'PROGRAM': 'aaa', '
BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+': 'aaa', '-': 'aaa', '/'
: 'aaa', '*': 'aaa', '(' : 'aaa', ')': 'aaa', ',': 'aaa', '.': 'aaa', ';': 'aaa', '=': 'aaa'
, ':': 'aaa', '$': 'aaa'},
        'U': {'P': 'P', 'Q': 'Q', 'R': 'R', 'S': 'S', '0': 'aaa', '1': 'aaa', '2': 'aaa', '
3': 'aaa', '4': 'aaa', '5': 'aaa', '6': 'aaa', '7': 'aaa', '8': 'aaa', '9': 'aaa', 'PROGRAM
': 'aaa', 'BEGIN': 'aaa', 'END.': 'aaa', 'INTEGER': 'aaa', 'PRINT': 'aaa', '+': 'aaa', '-'
: 'aaa', '/': 'aaa', '*': 'aaa', '(' : 'aaa', ')': 'aaa', ',': 'aaa', '.': 'aaa', ';': 'aaa'
, '=': 'aaa', ':': 'aaa', '$': 'aaa'},
    }

```

```

    parse_status = predictive_parser(token_list, predict_table, terminal_list,
starting_symbol='W')

```

```

    if parse_status is True:
        code_generator(content.split('\n'), 'main.cpp')

```

```

if __name__ == "__main__":
    main()

```