# AFLL Assignment

Name:- Ankit Bembalgi

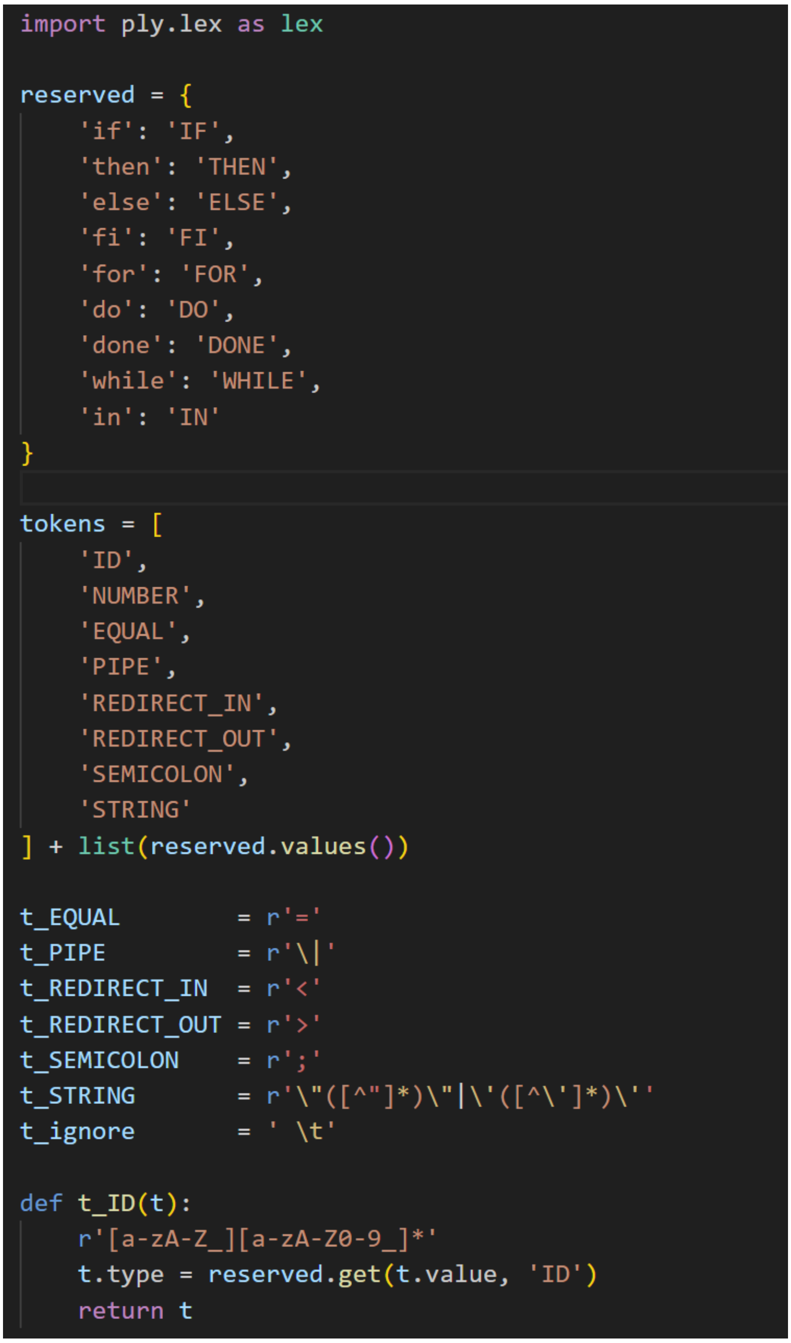
SRN:- PES2UG24AM020

Name:- Adarsh D

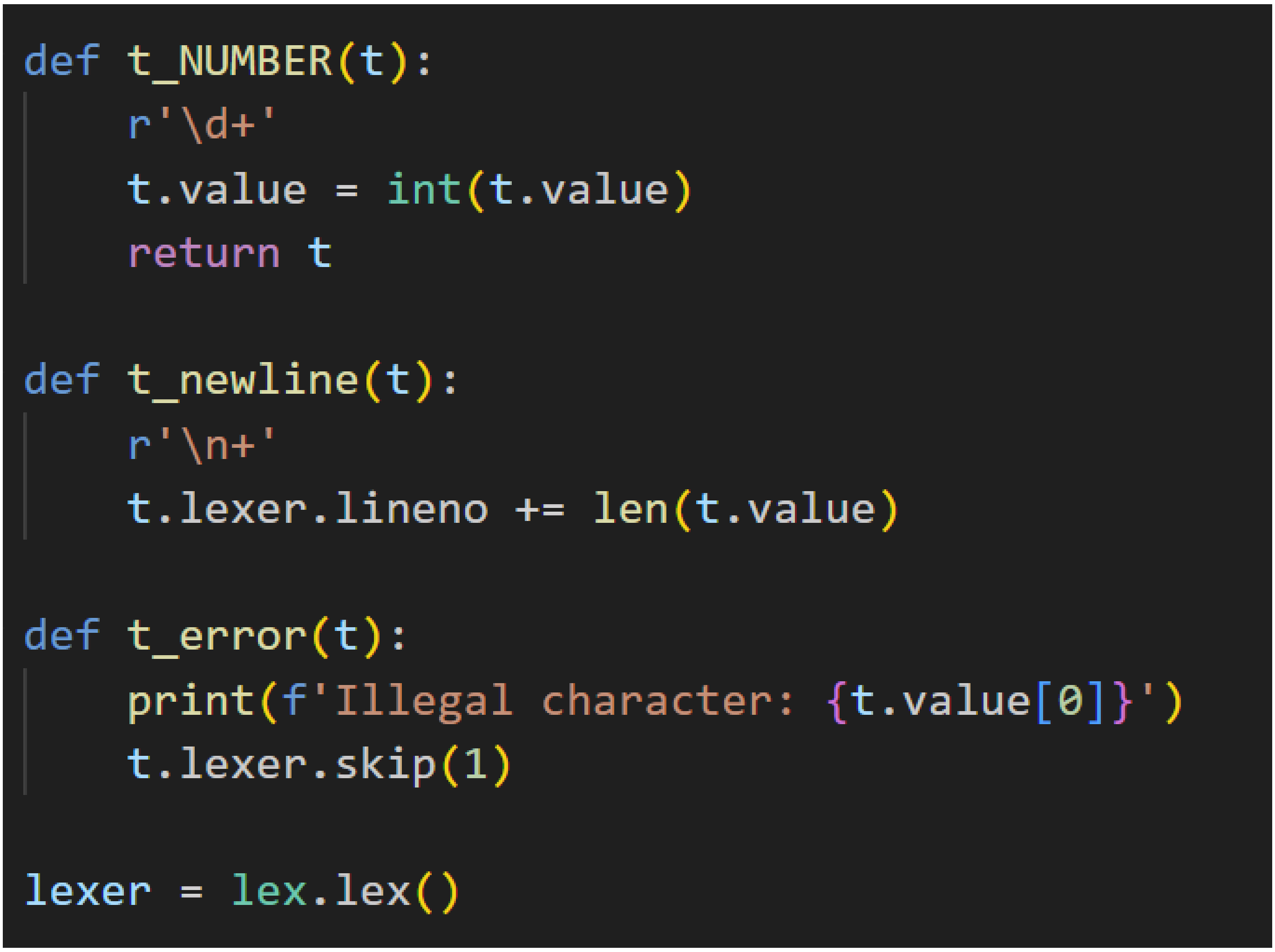
SRN:- PES2UG24AM010

Code:

1. Lexer.py

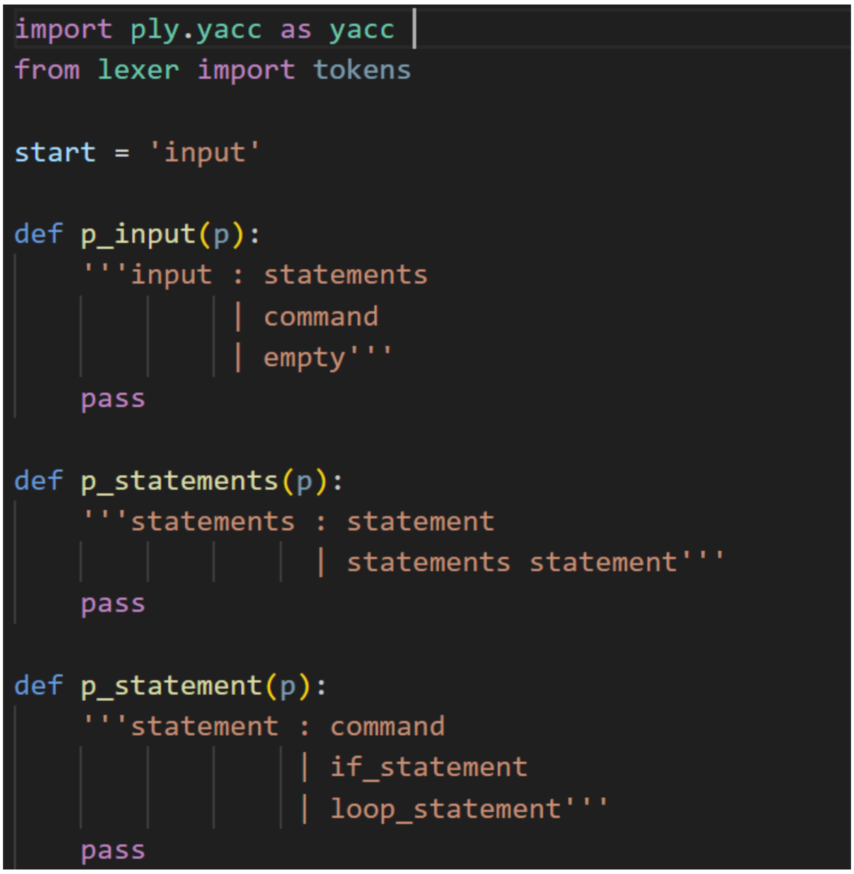


```python
import ply.lex as lex

reserved = {
    'if': 'IF',
    'then': 'THEN',
    'else': 'ELSE',
    'fi': 'FI',
    'for': 'FOR',
    'do': 'DO',
    'done': 'DONE',
    'while': 'WHILE',
    'in': 'IN'
}

tokens = [
    'ID',
    'NUMBER',
    'EQUAL',
    'PIPE',
    'REDIRECT_IN',
    'REDIRECT_OUT',
    'SEMICOLON',
    'STRING'
] + list(reserved.values())

t_EQUAL        = r'='
t_PIPE         = r'\|'
t_REDIRECT_IN  = r'<'
t_REDIRECT_OUT = r'>'
t_SEMICOLON    = r';'
t_STRING       = r'\"([^"]*)\"|\'([^\']*)\''
t_ignore       = ' \t'

def t_ID(t):
    r'[a-zA-Z_][a-zA-Z0-9_]*'
    t.type = reserved.get(t.value, 'ID')
    return t
```

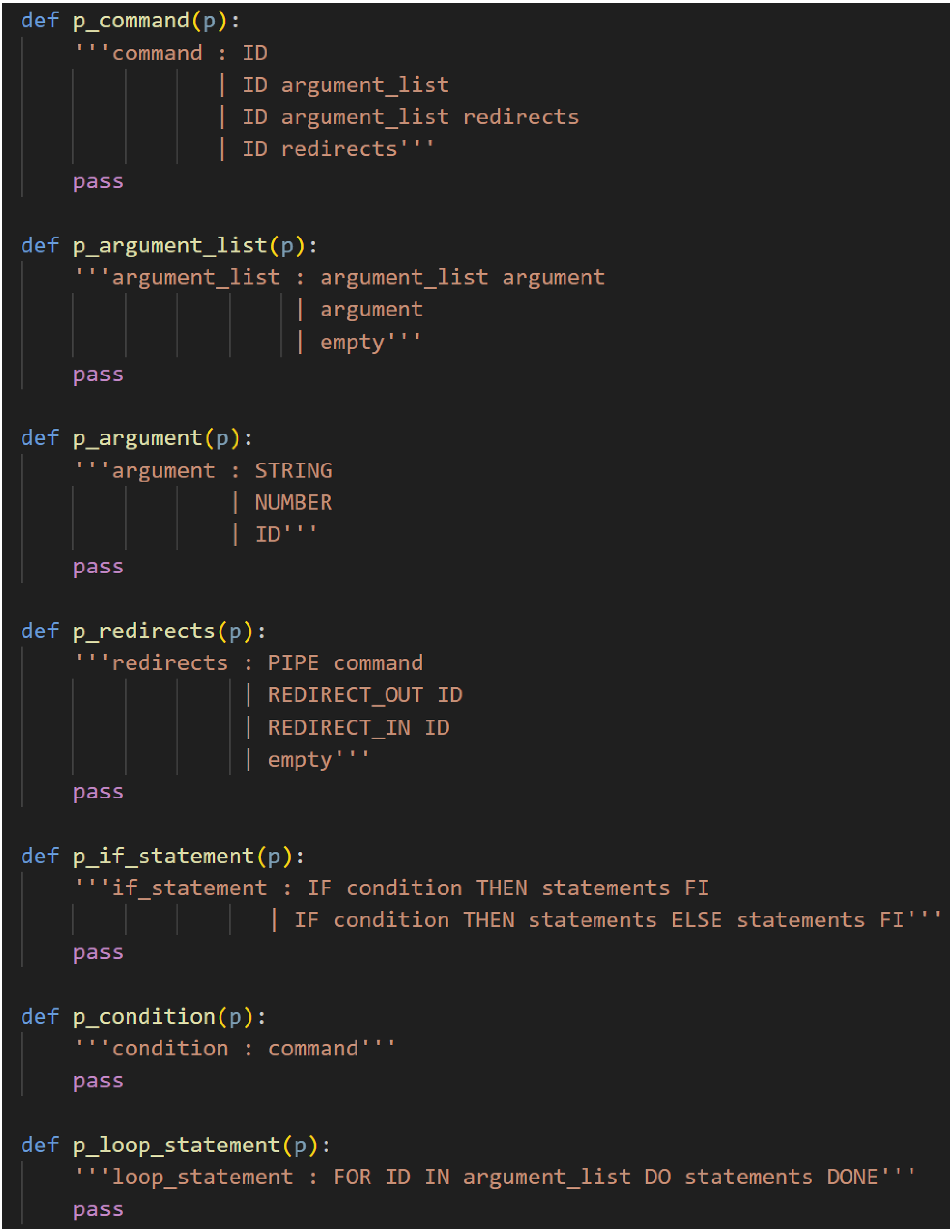


```python
def t_NUMBER(t):
    r'\d+'
    t.value = int(t.value)
    return t

def t_newline(t):
    r'\n+'
    t.lexer.lineno += len(t.value)

def t_error(t):
    print(f'Illegal character: {t.value[0]}')
    t.lexer.skip(1)

lexer = lex.lex()
```

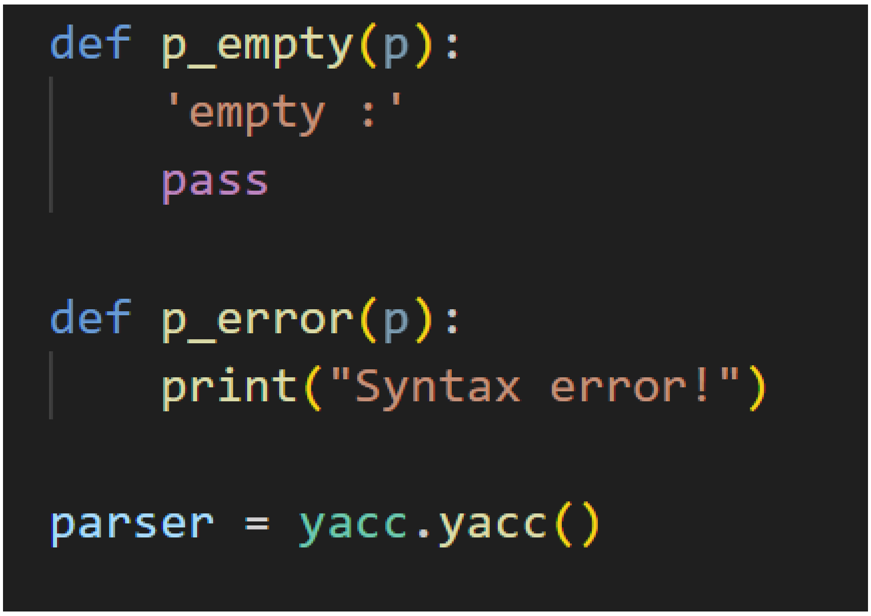
2. Parser.py



```python
import ply.yacc as yacc
from lexer import tokens

start = 'input'

def p_input(p):
    '''input : statements
             | command
             | empty'''
    pass

def p_statements(p):
    '''statements : statement
                  | statements statement'''
    pass

def p_statement(p):
    '''statement : command
                 | if_statement
                 | loop_statement'''
    pass
```

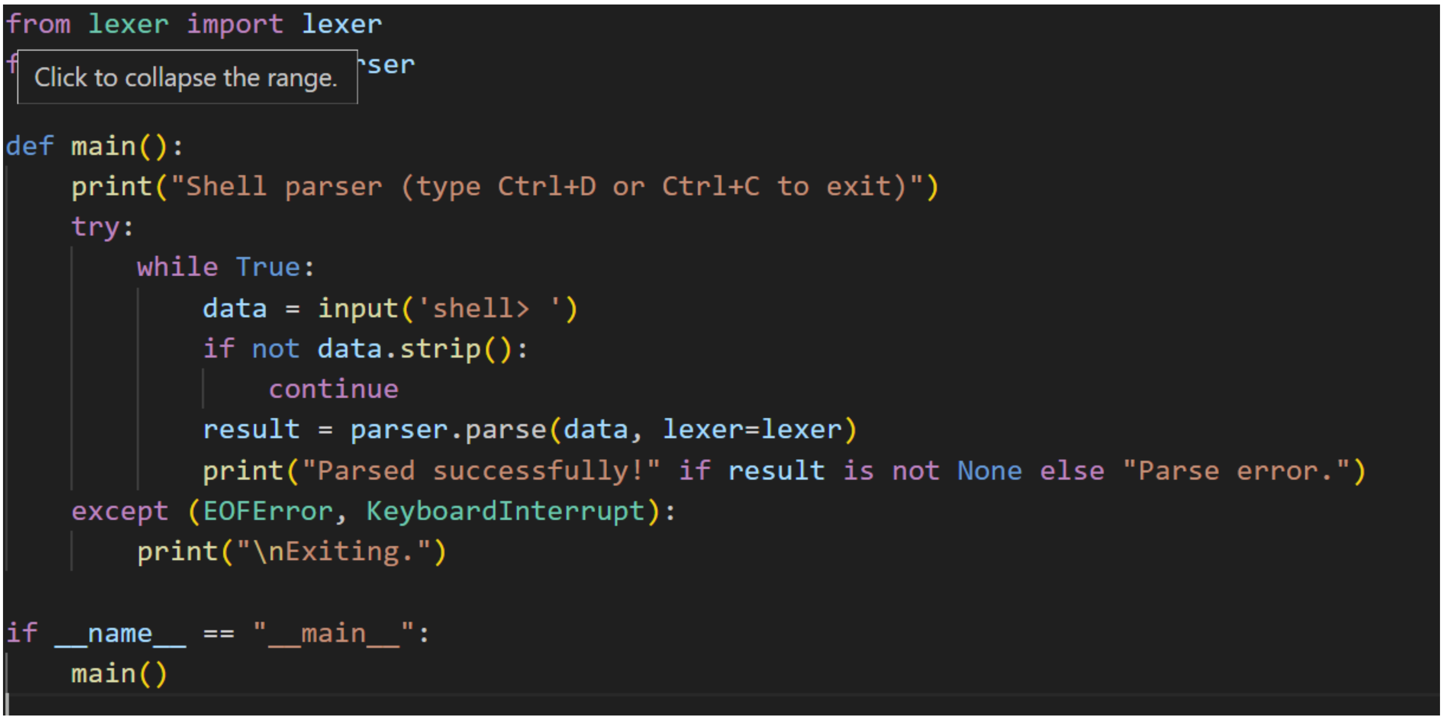


```
def p_command(p):
    '''command : ID
               | ID argument_list
               | ID argument_list redirects
               | ID redirects'''
    pass

def p_argument_list(p):
    '''argument_list : argument_list argument
                     | argument
                     | empty'''
    pass

def p_argument(p):
    '''argument : STRING
                | NUMBER
                | ID'''
    pass

def p_redirects(p):
    '''redirects : PIPE command
                 | REDIRECT_OUT ID
                 | REDIRECT_IN ID
                 | empty'''
    pass

def p_if_statement(p):
    '''if_statement : IF condition THEN statements FI
                   | IF condition THEN statements ELSE statements FI'''
    pass

def p_condition(p):
    '''condition : command'''
    pass

def p_loop_statement(p):
    '''loop_statement : FOR ID IN argument_list DO statements DONE'''
    pass
```



```python
def p_empty(p):
    'empty :'
    pass

def p_error(p):
    print("Syntax error!")

parser = yacc.yacc()
```

3. Main.py



```python
from lexer import lexer
f                    ser

def main():
    print("Shell parser (type Ctrl+D or Ctrl+C to exit)")
    try:
        while True:
            data = input('shell> ')
            if not data.strip():
                continue
            result = parser.parse(data, lexer=lexer)
            print("Parsed successfully!" if result is not None else "Parse error.")
    except (EOFError, KeyboardInterrupt):
        print("\nExiting.")

if __name__ == "__main__":
    main()
```