COMPILER DESIGN AS 5

```
import re

data_type={'int','float','char','string'}
regex = {
    'Keyword':
re.compile(r'\b(?:if|else|while|return|int|float|char|string|main|void)\b'),
    'Identifier': re.compile(r'\b[a-zA-Z_][a-zA-Z0-9_]*\b'),
    'Float': re.compile(r'\b\d+\.\d+\b'),
    'Int': re.compile(r'\b\d+\b'),
    'String': re.compile(r'"[^"]*"'),
    'Char': re.compile(r"'[^']'"),
    'Operator': re.compile(r'[+\-*/%==!=<>]=?|&&|\|\|'),
    'Delimiter': re.compile(r'[;,{}()]')
}
symbol_table, error_log = {}, []

def insert(var, type, lno):
    if var not in symbol_table:
        symbol_table[var] = {
            'type': type,
            'memory_location': hex(hash(id(var))),
            'line': lno }

def lexer():
    n = int(input("Enter your source code length: "))
    tokens, in_comment = [], False
    combined = ""

    for i in range(1,n+1):
        line = input()
        line = re.sub(r'//.*$', '', line)
        combined+=line
        if '/*' in line:
            in_comment = True
        if '*/' in line and in_comment:
            in_comment = False
            line = line.split('*/', 1)[1]
        if in_comment:
            continue
        line = line.strip()
        while line:
            matched = False
            for key, pattern in regex.items():
                match = pattern.match(line)
                if match:
                    variable = match.group(0)
                    var_type = key
                    if ((var_type == 'Keyword') and (variable in data_type)):
```

```
                        next_token = line[len(variable):].strip().split()[0] if
line[len(variable):].strip() else None
                        if next_token and regex['Identifier'].fullmatch(next_token):
                            insert(next_token, variable, i)

                    tokens.append((var_type, variable))
                    line = line[len(variable):].strip()
                    matched = True
                    break
            if not matched:
                error_log.append(f"Invalid character '{line[0]}' at line {i}")
                break

    if any(lit == "" for lit in re.findall(r'"(.*?)"', combined)):
        error_log.append("Unterminated string literal found.")
    if combined.count('/*') != combined.count('*/'):
        error_log.append("Unterminated multi-line comment found.")
    return tokens


tokens = lexer()
print("Tokens list\n","-"*60)
print(tokens)
print("-"*60)
print("\nSymbol Table:")
print(f"{'Identifier':<15}{'Data Type':<10}{'Memory Address':<20}{'Line number'}")
for sym, info in symbol_table.items():
    print(f"{sym:<15}{info['type']:<10}{info['memory_location']:<20}{info['line']}")

print("\nErrors:")
if error_log:
    with open("errorlog.log","w") as f:
        f.write("\n".join(error_log))
        print("Error Successfully logged in errorlog.log")
else:
    print("No errors found.")
```

TEST CASE 1 :

```
= RESTART: C:\Users\Samar Mittal\Desktop\Compiler LAb\lab5\q1.py
Enter your source code length: 7
int main(){
int x = 34;
float y = 23.234;
char k = 'S' ;
char #4z = 'A' ;
string sam ="Samar Mittal";
}
Tokens list
 ------------------------------------------------------------
[('Keyword', 'int'), ('Keyword', 'main'), ('Delimiter', '('), ('Delimiter', ')')
, ('Delimiter', '{'), ('Keyword', 'int'), ('Identifier', 'x'), ('Operator', '=')
, ('Int', '34'), ('Delimiter', ';'), ('Keyword', 'float'), ('Identifier', 'y'),
('Operator', '='), ('Float', '23.234'), ('Delimiter', ';'), ('Keyword', 'char'),
 ('Identifier', 'k'), ('Operator', '='), ('Char', "'S'"), ('Delimiter', ';'), ('
Keyword', 'char'), ('Keyword', 'string'), ('Identifier', 'sam'), ('Operator', '=
'), ('String', '"Samar Mittal"'), ('Delimiter', ';'), ('Delimiter', '}')]
------------------------------------------------------------

Symbol Table:
Identifier     Data Type Memory Address      Line number
x              int       0x7ff91b7ef558      2
y              float     0x7ff91b7ef588      3
k              char      0x7ff91b7ef2e8      4
sam            string    0x1be211663d0       6

Errors:
Error Successfully logged in errorlog.log
```

errorlog.log - Notepad

File Edit Format View Help

```
Invalid character '#' at line 5
```

TEST CASE 2 :

```
= RESTART: C:\Users\Samar Mittal\Desktop\Compiler LAb\lab5\q1.py
Enter your source code length: 2
char c ='A' ;
string /*
Tokens list
 ------------------------------------------------------------
[('Keyword', 'char'), ('Identifier', 'c'), ('Operator', '='), ('Char', "'A'")
, ('Delimiter', ';')]
------------------------------------------------------------

Symbol Table:
Identifier     Data Type Memory Address      Line number
c              char      0x7ff91b7ef168      1

Errors:
Error Successfully logged in errorlog.log
```

errorlog.log - Notepad

File Edit Format View Help

```
Unterminated multi-line comment found.
```