Test cases(Tokenization) :

```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gc
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
while (x < 10)
    x++;
    printf("X is now %d", x);
^D
Lexical Analysis:
Token: while, Type: Keyword
Token: (, Type: Left Parenthesis
Token: x, Type: Identifier
Token: <, Type: Comparison Operator
Token: 10, Type: Number
Token: ), Type: Right Parenthesis
Token: x, Type: Identifier
Token: +, Type: Operator
Token: +, Type: Operator
Token: ;, Type: Semicolon
Token: printf, Type: Identifier
Token: (, Type: Left Parenthesis
Token: "X is now %d", Type: String Literal
Token: ,, Type: Comma
Token: x, Type: Identifier
Token: ), Type: Right Parenthesis
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
```

```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gc
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
int @c=10;
^D
Lexical Analysis:
Token: int, Type: Keyword
Unknown token: @
Token: c, Type: Identifier
Token: =, Type: Assignment Operator
Token: 10, Type: Number
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
```

```
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
int $c=10;
^Z

Lexical Analysis:
Token: int, Type: Keyword
Unknown token: $
Token: c, Type: Identifier
Token: =, Type: Assignment Operator
Token: 10, Type: Number
Token: ;, Type: Semicolon
PS C:\Users\uditi\AppData\Local\Temp\871d0649-cdd8-4db0-89c6-9446b3fe46fb_Compiler-Design-maste
> nt
```

```
}
^D
Lexical Analysis:
Token: int, Type: Keyword
Token: main, Type: Identifier
Token: (, Type: Left Parenthesis
Token: ), Type: Right Parenthesis
Token: {, Type: Left Brace
Token: int, Type: Keyword
Token: x, Type: Identifier
Token: =, Type: Assignment Operator
Token: 5, Type: Number
Token: ;, Type: Semicolon
Token: while, Type: Keyword
Token: (, Type: Left Parenthesis
Token: x, Type: Identifier
Token: >, Type: Comparison Operator
Token: 0, Type: Number
Unknown token: @
Token: ), Type: Right Parenthesis
Token: {, Type: Left Brace
Token: /, Type: Operator
Token: /, Type: Operator
Token: Invalid, Type: Identifier
Token: character, Type: Identifier
Unknown token: '
Unknown token: @
Unknown token: '
Token: in, Type: Identifier
Token: condition, Type: Identifier
Token: x, Type: Identifier
Token: -, Type: Operator
Token: -, Type: Operator
Token: ;, Type: Semicolon
Token: }, Type: Right Brace
Token: return, Type: Keyword
Token: 0, Type: Number
Token: ;, Type: Semicolon
Token: }, Type: Right Brace
ayushi@Ayushis-MacBook-Air 0 %
```

Ln

```
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
int main() {
    int x = 5;
    while (x > 0 @) {  // Invalid character '@' in condition
        x--;
    }
    return 0;
}
^D
Lexical Analysis:
Token: int, Type: Keyword
Token: main, Type: Identifier
Token: (, Type: Left Parenthesis
Token: ), Type: Right Parenthesis
Token: {, Type: Left Brace
Token: int, Type: Keyword
Token: x, Type: Identifier
Token: =, Type: Assignment Operator
Token: 5, Type: Number
Token: ;, Type: Semicolon
Token: while, Type: Keyword
Token: (, Type: Left Parenthesis
Token: x, Type: Identifier
Token: >, Type: Comparison Operator
Token: 0, Type: Number
Unknown token: @
Token: ), Type: Right Parenthesis
Token: {, Type: Left Brace
Token: /, Type: Operator
Token: /, Type: Operator
Token: Invalid, Type: Identifier
Token: character, Type: Identifier
Unknown token: '
Unknown token: @
Unknown token: '
Token: in, Type: Identifier
Token: condition, Type: Identifier
Token: x, Type: Identifier
Token: -, Type: Operator
Token: -, Type: Operator
```

```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gcc lexi
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
int 1stValue = 10;
float @price = 15.5;
^D
Lexical Analysis:
Token: int, Type: Keyword
Token: 1, Type: Number
Token: stValue, Type: Identifier
Token: =, Type: Assignment Operator
Token: 10, Type: Number
Token: ;, Type: Semicolon
Token: float, Type: Keyword
Unknown token: @
Token: price, Type: Identifier
Token: =, Type: Assignment Operator
Token: 15.5, Type: Number
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
```

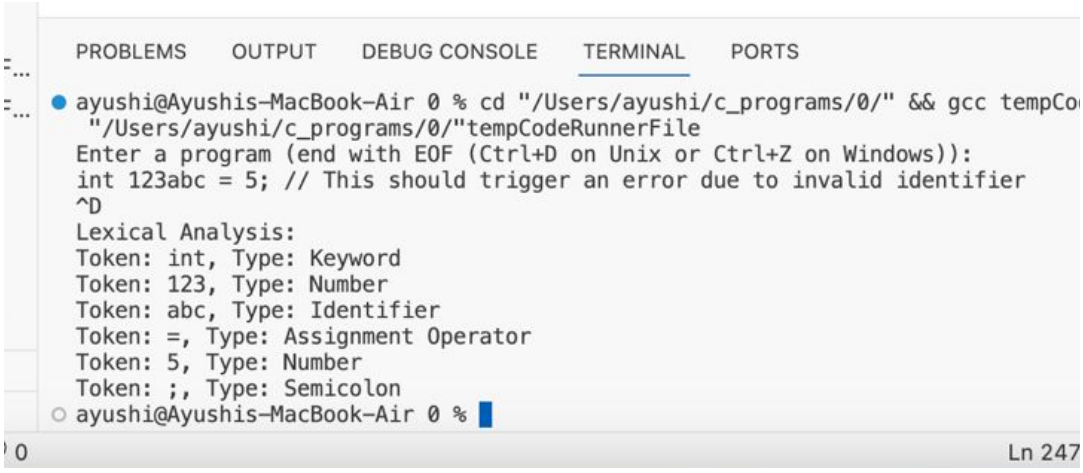
```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gcc lex
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
int 1stVariable = 10;
^D
Lexical Analysis:
Token: int, Type: Keyword
Token: 1, Type: Number
Token: stVariable, Type: Identifier
Token: =, Type: Assignment Operator
Token: 10, Type: Number
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
```

```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gcc l
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
float $price = 20.5;
^D
Lexical Analysis:
Token: float, Type: Keyword
Unknown token: $
Token: price, Type: Identifier
Token: =, Type: Assignment Operator
Token: 20.5, Type: Number
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
```

```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gcc
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
char @letter = 'A';
^D
Lexical Analysis:
Token: char, Type: Keyword
Unknown token: @
Token: letter, Type: Identifier
Token: =, Type: Assignment Operator
Unknown token: '
Token: A, Type: Identifier
Unknown token: '
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
```

```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gcc lexical.
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
if (x $= 5)
^D
Lexical Analysis:
Token: if, Type: Keyword
Token: (, Type: Left Parenthesis
Token: x, Type: Identifier
Unknown token: $
Token: =, Type: Assignment Operator
Token: 5, Type: Number
Token: ), Type: Right Parenthesis
ayushi@Ayushis-MacBook-Air 0 %
```

```
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gcc
rams/0/"lexical
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows):
int total = a + b *;
^D
Lexical Analysis:
Token: int, Type: Keyword
Token: total, Type: Identifier
Token: =, Type: Assignment Operator
Token: a, Type: Identifier
Token: +, Type: Operator
Token: b, Type: Identifier
Token: *, Type: Operator
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
ayushi@Ayushis-MacBook-Air 0 % cd "/Users/ayushi/c_programs/0/" && gcc tempCo
"/Users/ayushi/c_programs/0/"tempCodeRunnerFile
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows)):
int 123abc = 5; // This should trigger an error due to invalid identifier
^D
Lexical Analysis:
Token: int, Type: Keyword
Token: 123, Type: Number
Token: abc, Type: Identifier
Token: =, Type: Assignment Operator
Token: 5, Type: Number
Token: ;, Type: Semicolon
ayushi@Ayushis-MacBook-Air 0 %
0
Ln 247


"/Users/ayushi/c_programs/0/"tempCodeRunnerFile
Enter a program (end with EOF (Ctrl+D on Unix or Ctrl+Z on Windows)):
for (int i = 0; i < n; i++) {
if (arr[i] > 0) {
positiveCount++;
}
}
^D
Lexical Analysis:
Token: for, Type: Keyword
Token: (, Type: Left Parenthesis
Token: int, Type: Keyword
Token: i, Type: Identifier
Token: =, Type: Assignment Operator
Token: 0, Type: Number
Token: ;, Type: Semicolon
Token: i, Type: Identifier
Token: <, Type: Comparison Operator
Token: n, Type: Identifier
Token: ;, Type: Semicolon
Token: i, Type: Identifier
Token: +, Type: Operator
Token: +, Type: Operator
Token:), Type: Right Parenthesis
Token: {, Type: Left Brace
Token: if, Type: Keyword
Token: (, Type: Left Parenthesis
Token: arr, Type: Identifier
Token: [, Type: Left Bracket
Token: i, Type: Identifier
Token:], Type: Right Bracket
Token: >, Type: Comparison Operator
Token: 0, Type: Number
Token:), Type: Right Parenthesis
Token: {, Type: Left Brace
Token: positiveCount, Type: Identifier
Token: +, Type: Operator
Token: +, Type: Operator
Token: ;, Type: Semicolon
Token: }, Type: Right Brace
0
Ln