

# Project -1

## Programming Languages

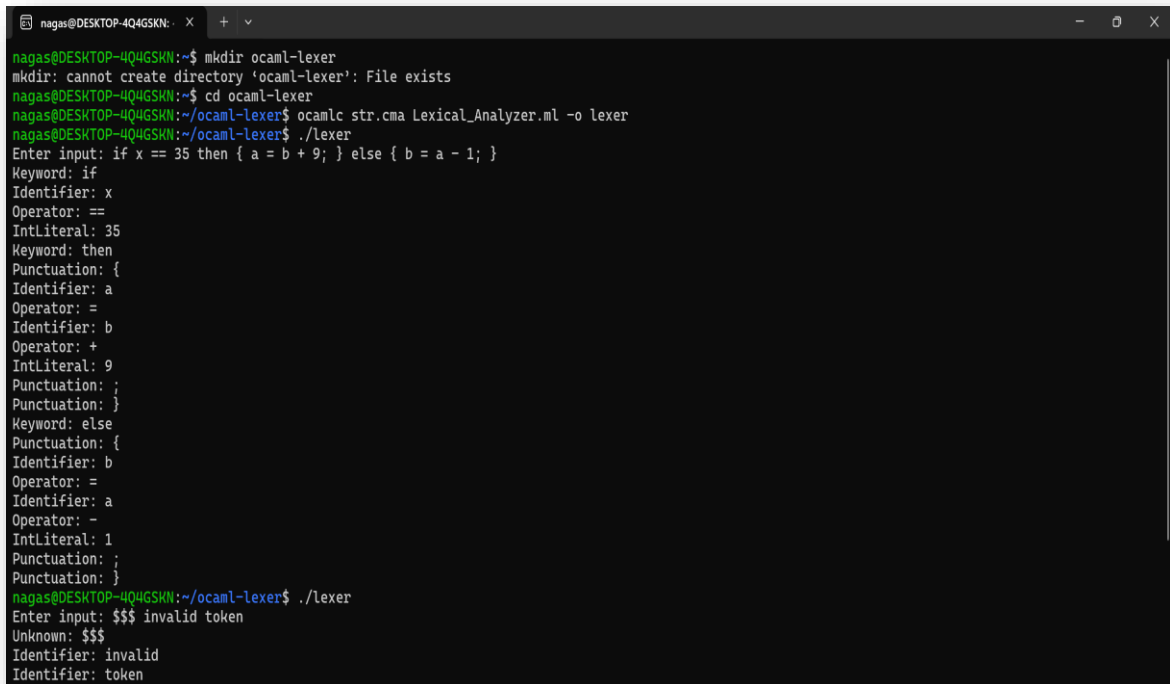
### LEXICAL ANALYZER

The output of the Lexical Analyzer:

In this first Screenshot we gave the

Test case 1: “if x == 35 then { a = b + 9; } else { b = a - 1; }”

Test case 2: “\$\$\$ invalid token”



```
nagas@DESKTOP-4Q4GSKN: ~$ mkdir ocaml-lexer
mkdir: cannot create directory 'ocaml-lexer': File exists
nagas@DESKTOP-4Q4GSKN: ~$ cd ocaml-lexer
nagas@DESKTOP-4Q4GSKN: ~/ocaml-lexer$ ocamlc str.cma Lexical_Analyzer.ml -o lexer
nagas@DESKTOP-4Q4GSKN: ~/ocaml-lexer$ ./lexer
Enter input: if x == 35 then { a = b + 9; } else { b = a - 1; }
Keyword: if
Identifier: x
Operator: ==
IntLiteral: 35
Keyword: then
Punctuation: {
Identifier: a
Operator: =
Identifier: b
Operator: +
IntLiteral: 9
Punctuation: ;
Punctuation: }
Keyword: else
Punctuation: {
Identifier: b
Operator: =
Identifier: a
Operator: -
IntLiteral: 1
Punctuation: ;
Punctuation: }
nagas@DESKTOP-4Q4GSKN: ~/ocaml-lexer$ ./lexer
Enter input: $$$ invalid token
Unknown: $$$
Identifier: invalid
Identifier: token
```

And I also gave some other inputs to check the lexical analyser is working properly or not.

The inputs given are -

1.  $x = y - 5;$
2.  $\text{if } y \neq 20 \text{ then } \{ z = x * 3; \} \text{ else } \{ x = z / 2; \}$

```
nagas@DESKTOP-4Q4GSKN:~/ocaml-lexer$ ./lexer
Enter input: x = y - 5;
Identifier: x
Operator: =
Identifier: y
Operator: -
IntLiteral: 5
Punctuation: ;
nagas@DESKTOP-4Q4GSKN:~/ocaml-lexer$ ./lexer
Enter input: if y != 20 then { z = x * 3; } else { x = z / 2; }
Keyword: if
Identifier: y
Operator: !=
IntLiteral: 20
Keyword: then
Punctuation: {
Identifier: z
Operator: =
Identifier: x
Operator: *
IntLiteral: 3
Punctuation: ;
Punctuation: }
Keyword: else
Punctuation: {
Identifier: x
Operator: =
Identifier: z
Operator: /
IntLiteral: 2
Punctuation: ;
Punctuation: }
nagas@DESKTOP-4Q4GSKN:~/ocaml-lexer$
```