



NOTRE DAME UNIVERSITY
BANGLADESH

CSE-3104 LAB Report-04

Course Title: Compiler Design Lab

Course Code: CSE-3104

Submitted by:

Name: Istiak Alam

ID: 0692230005101005

Batch: CSE-20

Submission Date: 24-11-24

Lab Task Topic: Compiler Problem Solving using Lex

Submitted to:

Khorshed Alam

Lecturer, NDUB

Problem:

Write a YACC Program to implement if-then-else statement.

Solution:

Code : –

```
// Lex File -> ifelse.l //
%{
#include "ifelse.tab.h"
%}
alpha [A-Za-z]
digit [0-9]
%%
[\\t\\n]
if      return IF;
then    return THEN;
else    return ELSE;
{digit}+({alpha}|{digit})*    return ID;
"<="    return LE;
">="    return GE;
"=="    return EQ;
"!="    return NE;
"||"    return OR;
"&&"    return AND;
.       return yytext[0];
%%

// Yacc File -> ifelse.y //
%{
#include<stdio.h>
#include<stdlib.h>
extern int yylex();
void yyerror(const char *s);
%}
%token ID NUM IF THEN LE GE EQ NE OR AND ELSE
%right '='
%left AND OR
%left '<'>' LE GE EQ NE
%left '+' '-'
%left '*' '/'
%right UMINUS ac
%left '|'
%
S : ST{printf("Input Accepted.\\n");exit(0);};
ST : IF '(' E2 ')' THEN ST1 ';' ELSE ST1 ';'
    | IF '(' E2 ')' THEN ST1 ';'
    ;

ST1 : ST
    | E
    ;
E : ID '=' E
  | E '+' E
  | E '-' E
  | E '*' E
  | E '/' E
  | E '<' E
  | E '>' E
  | E LE E
  | E GE E
  | E EQ E
  | E NE E
  | E OR E
  | E AND E
  | ID|NUM;
```

```

E2:      E '<' E
         | E '>' E
         | E LE E
         | E GE E
         | E EQ E
         | E NE E
         | E OR E
         | E AND E
         | ID
         | NUM
         ;

%%
void yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
}
int main() {
    printf("Enter The Expression: ");
    yyparse();
    return 0;
}

```

Code Processing –

Step 1 : Open terminal in Linux and create a lex file named **ifelse.l**

⇒ **touch ifelse.l**

Step 1 : Open terminal in Linux and create a yacc file named **ifelse.y**

⇒ **touch ifelse.y**

Step 2 : After Writing those code save it and type command in terminal :

⇒ **bison -d ifelse.y**

⇒ **lex ifelse.l**

Step 3 : It will create a lex.yy.c & yyfile. Then next type command in terminal

⇒ **gcc -o IFELSE ifelse.tab.c lex.yy.c -lfl -lm**

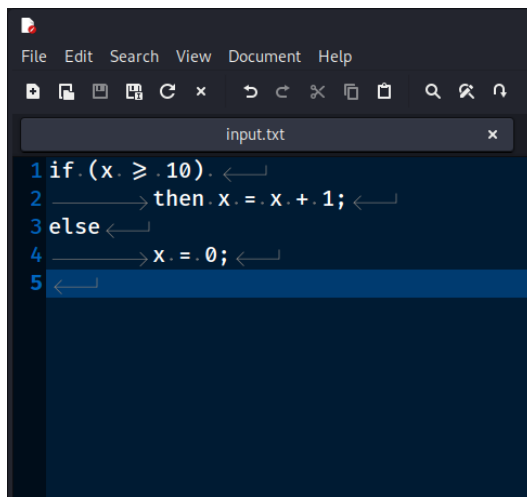
Step 4 : Then it will create an executable file named **IFELSE.exe**.

Step 5 : Next type command in terminal and run the exe file :

⇒ **./IFELSE < input.txt**

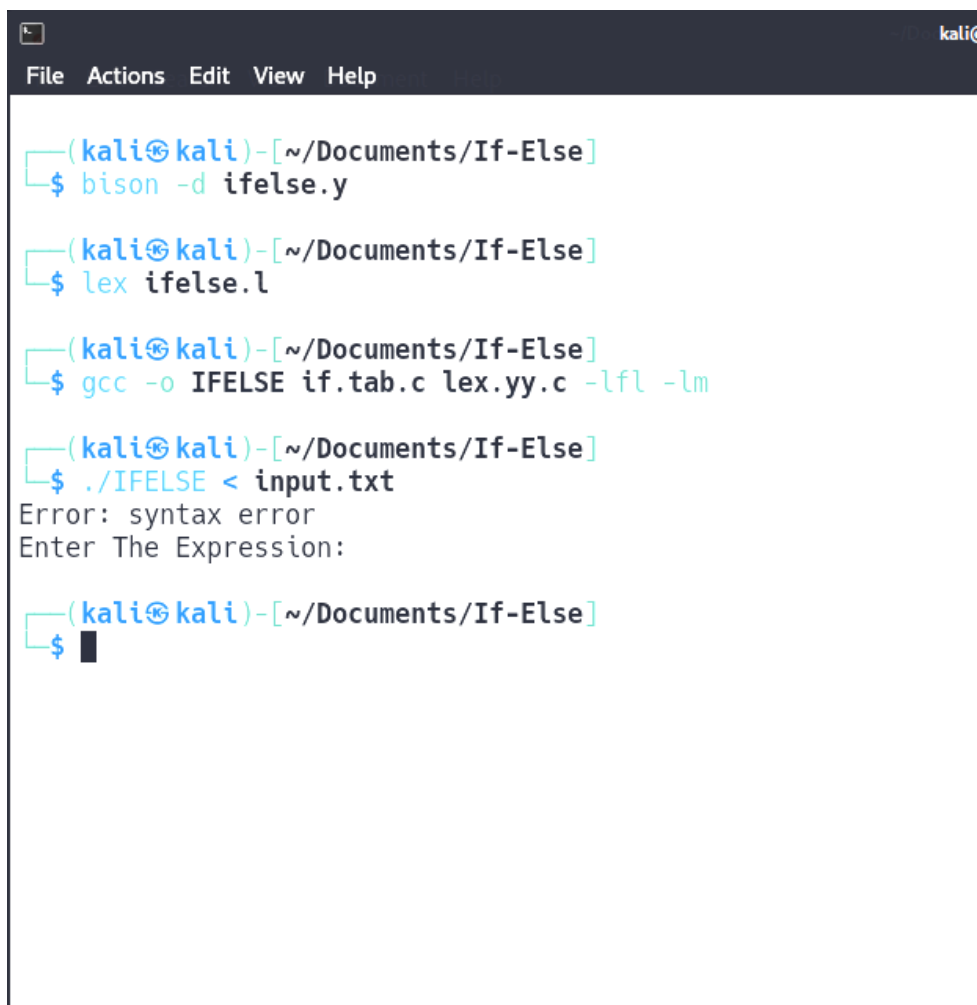
Input and Output:

1. We are using **input.txt** file for checking out inputs in this lexical analyzer.
2. In the **input.txt** file there are a simple program language –



```
File Edit Search View Document Help
input.txt
1 if (x ≥ 10)
2     then x = x + 1;
3 else
4     x = 0;
5
```

3. After injecting the input, the output is –



```
kali
File Actions Edit View Help
(kali@kali)-[~/Documents/If-Else]
└─$ bison -d ifelse.y
(kali@kali)-[~/Documents/If-Else]
└─$ lex ifelse.l
(kali@kali)-[~/Documents/If-Else]
└─$ gcc -o IFELSE if.tab.c lex.yy.c -lfl -lm
(kali@kali)-[~/Documents/If-Else]
└─$ ./IFELSE < input.txt
Error: syntax error
Enter The Expression:
(kali@kali)-[~/Documents/If-Else]
└─$
```