Exp 7d : YACC – IF ELSE

LEX

```
1. Start
2. %% - rule section
       "if" return IF
       "else" return ELSE
       [ \t\n]
       [0-9]+ return NUM
       [a-zA-Z][a-zA-Z0-9]*  return ID
       \"[^\"]*\"  return STRING
       "<"  return L
       ">"  return G
       "<="  return LE
       ">="  return GE
       "=="  return EE
       "!="  return NE
       "++"  return INC
       "--"  return DEC
       "||"  return OR
       "&&"  return AND
       .  return yytext[0]
3. yywrap() retturn 1
4. Stop
```

YACC

```
1. Start
2. %token IF ELSE L G LE GE EE NE INC DEC OR AND ID NUM STRING
3. %% rule section
       S : if else { print "valid if else statement" } ;
       if : IF '(' cond ')' '{' stmt '}' ;
       else : ELSE '{' stmt '}' | ;
       cond : scond | scond AND cond | scond OR cond ;
       scond : nid | nid relop nid ;
       nid : ID | NUM ;
       relop : L | G | LE | GE | EE | NE ;
       stmt : ID '(' STRING other ')' ';' stmt | E ';' stmt | ;
       other : ',' ID other | ',' '&' ID other | ;
       E : ID'='E | E'+'E | E'-'E | E'*'E | E'/'E | E INC | E DEC | nid | '(' nid ')' ;
4. yyerror() to handle error
5. in main() call yyparse()
```

if else – Lex

```
%{
   #include<stdio.h>
   #include "y.tab.h"
%}

%%
"if" { return IF; }
"else" { return ELSE; }
```

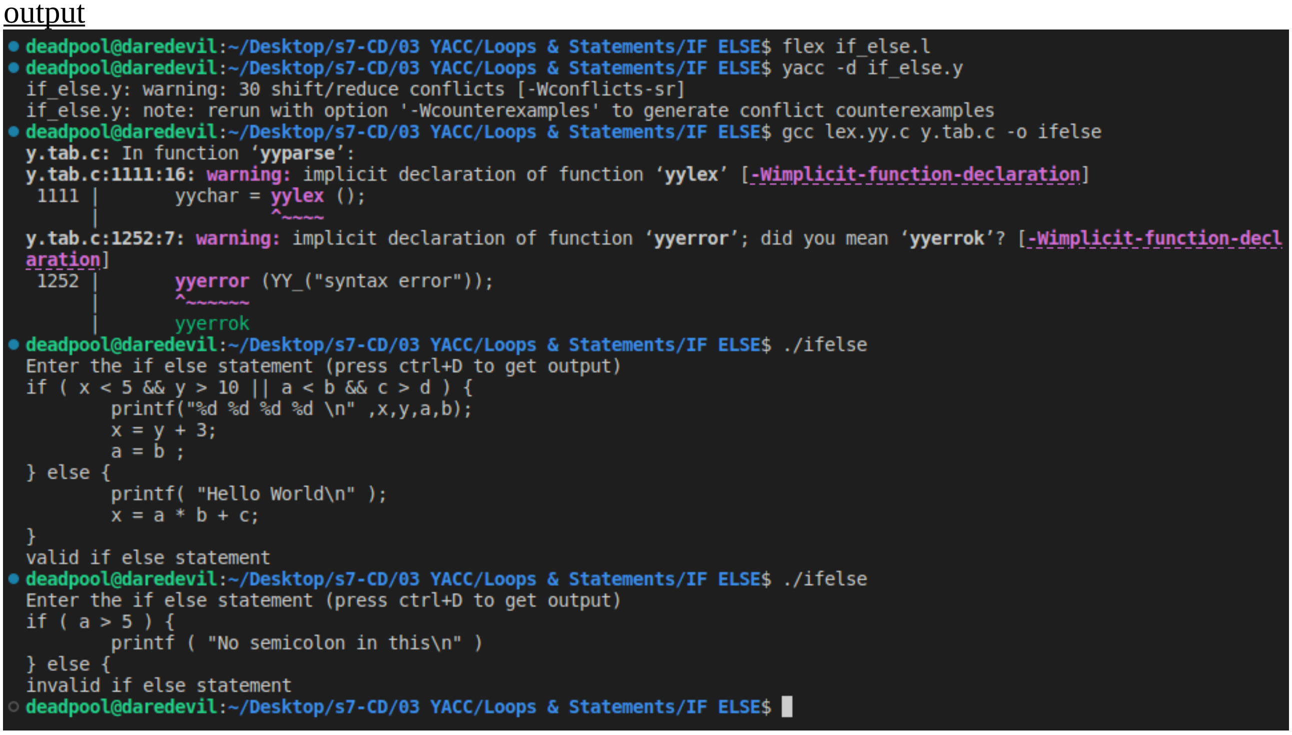
```
[ \t\n]
[0-9]+ { return NUM; }
[a-zA-Z][a-zA-Z0-9]* { return ID; }
\"[^\"]*\" { return STRING; }

"<" { return L; }
">" { return G; }
"<=" { return LE; }
">=" { return GE; }
"==" { return EE; }
"!=" { return NE; }
"++" { return INC; }
"--" { return DEC; }
"||" { return OR; }
"&&" { return AND; }
. { return yytext[0]; }
%%

int yywrap(){
   return 1;
}
```

if else – YACC

```
%{
   #include<stdio.h>
%}

%token IF ELSE L G LE GE EE NE INC DEC OR AND ID NUM STRING

%%
S : if else { printf("valid if else statement\n"); };
if : IF '(' cond ')' '{' stmt '}' ;
else : ELSE '{' stmt '}' | ;

cond : scond | scond AND cond | scond OR cond ;
scond : nid | nid relop nid ;
nid : ID | NUM ;
relop : L | G | LE | GE | EE | NE ;
stmt : ID '(' STRING other ')' ';' stmt | E ';' stmt | ;
other : ',' ID other | ',' '&' ID other | ;

E : ID'='E
| E'+'E
| E'-'E
| E'*'E
| E'/'E
| E INC
| E DEC
| nid
| '(' nid ')'
;
```

```
%%

int yyerror(){
    printf("invalid if else statement\n");
    return 1;
}

int main(){
    printf("Enter the if else statement (press ctrl+D to get output)\n");
    yyparse();
    return 0;
}
```

<u>output</u>



```
● deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/IF ELSE$ flex if_else.l
● deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/IF ELSE$ yacc -d if_else.y
  if_else.y: warning: 30 shift/reduce conflicts [-Wconflicts-sr]
  if_else.y: note: rerun with option '-Wcounterexamples' to generate conflict counterexamples
● deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/IF ELSE$ gcc lex.yy.c y.tab.c -o ifelse
  y.tab.c: In function ‘yyparse’:
  y.tab.c:1111:16: warning: implicit declaration of function ‘yylex’ [-Wimplicit-function-declaration]
   1111 |       yychar = yylex ();
        |                ^~~~~
  y.tab.c:1252:7: warning: implicit declaration of function ‘yyerror’; did you mean ‘yyerrok’? [-Wimplicit-function-decl
  aration]
   1252 |       yyerror (YY_("syntax error"));
        |       ^~~~~~~
        |       yyerrok
● deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/IF ELSE$ ./ifelse
  Enter the if else statement (press ctrl+D to get output)
  if ( x < 5 && y > 10 || a < b && c > d ) {
          printf("%d %d %d %d \n" ,x,y,a,b);
          x = y + 3;
          a = b ;
  } else {
          printf( "Hello World\n" );
          x = a * b + c;
  }
  valid if else statement
● deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/IF ELSE$ ./ifelse
  Enter the if else statement (press ctrl+D to get output)
  if ( a > 5 ) {
          printf ( "No semicolon in this\n" )
  } else {
  invalid if else statement
○ deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/IF ELSE$ █
```