

## Week-10

**Implement LALR parser using LEX and YACC for the following Grammar:**

$E \rightarrow E+T \mid T$

$E' \rightarrow T * F \mid F$

$F \rightarrow (E) \mid d$

**Code:**

**Parser.y file code:**

```
%{
#include<stdio.h>
%}
%token NUMBER
%%
S: E { printf("The Result is %d", $1); }
;
E: E'+T { $$ = $1 + $3; }
| T { $$ = $1; }
;
T: T'*F { $$ = $1 * $3; }
| F { $$ = $1; }
;
F: '{E}' { $$ = $2; }
| NUMBER { $$ = $1; }
;
%%
int main(){
  yyparse();
}
int yywrap() {
  return 1;
}
```

```
}  
void yyerror(char *s) {  
printf("Error %s", s);  
}
```

### **Parser.l file code:**

```
%{  
#include "parser.tab.h"  
%}  
%%  
[0-9]+ { yylval = atoi(yytext);  
return NUMBER;  
}  
[\t];  
\n return 0;  
. return yytext[0];  
%%
```

### **Output:**

```
C:\Users\Yeswanth\Downloads\Sem5\Compilers\CD-Lex-Programs>flex parser.l  
C:\Users\Yeswanth\Downloads\Sem5\Compilers\CD-Lex-Programs>yacc -d parser.y  
C:\Users\Yeswanth\Downloads\Sem5\Compilers\CD-Lex-Programs>gcc lex.yy.c parser.tab.c -w  
C:\Users\Yeswanth\Downloads\Sem5\Compilers\CD-Lex-Programs>a  
3+3*3  
The Result is 12
```