Exp No: 5

# IMPLEMENTATION OF CALCULATOR IN YACC AND LEX

Date: 09/03/2021

Name: MADHUMITHA S

Register Number: 185001086

---

1) **Aim:**
   To implement a calculator that implements arithmetic, relational, Boolean and conditional computations on integer data using YACC and LEX tool preserving associativity and precedence rules.

**2) PROGRAM CODE:**

**LEX PROGRAM**

```
%{
    #include<stdio.h>
    #include<string.h>
    #include<stdlib.h>
    #include "calculator.tab.h"
    void yyerror(char*);
    extern int yylval;

%}


%%
[\t]+;
[0-9]+ {yylval=atoi(yytext);return INTEGER;}
"+"|"-"|"/"|"*"|"&"|"|"|"<<"|">>"|"=="|"!="|"?"|":"|"^" {return *yytext;}
"("|")"|"\n" {return *yytext;}
. {char msg[25];sprintf(msg,"%s<%s>","invalid character",yytext);yyerror(msg);
}

%%
```

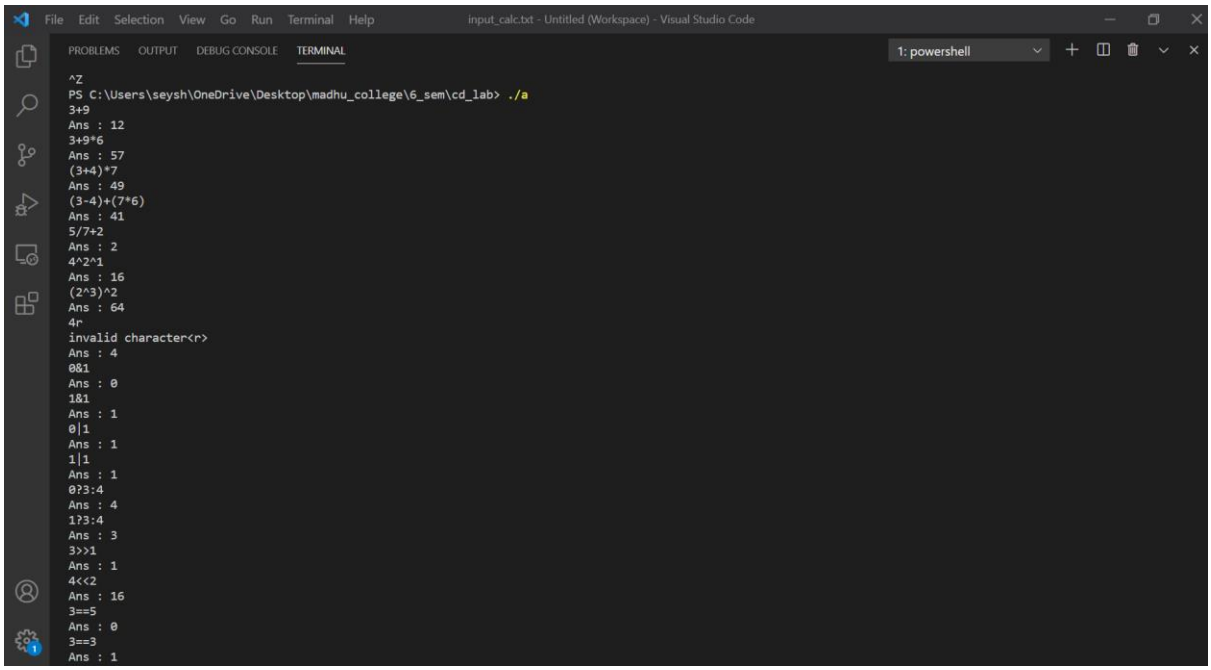
**YACC PROGRAM**

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

```
    #include<stdlib.h>
   #include "calculator.tab.h"
    #include<math.h>
    int yylex(void);

%}
%token INTEGER
%%
program : line program
        | line
line : expr '\n' {printf("Ans : %d\n",$1);}
expr : expr '+' mulex {$$=$1+$3;}
     | expr '-' mulex {$$=$1-$3;}
     | mulex {$$=$1;}
mulex : mulex '*' expoex {$$=$1*$3;}
      | mulex '/' expoex {$$=$1/$3;}
      | expoex {$$=$1;}
expoex : expoex '^' term {$$=pow($1,$3);}
       | boolex {$$=$1;}
boolex : boolex '&' boolex {$$=$1&$3;}
       | boolex '|' boolex {$$=$1|$3;}
       | bitex {$$=$1;}
bitex : bitex '<<' bitex {$$=$1<<$3;}
      | bitex '>>' bitex {$$=$1>>$3;}
      | relex {$$=$1;}
relex : relex '==' relex {$$=($1==$3);}
      | relex '!=' relex {$$=($1!=$3);}
      | condex {$$=$1;}
condex : condex '?' condex ':' condex {$$=$1?$3:$5;}
       | term {$$=$1;}
term : '(' expr ')' {$$=$2;}
     | INTEGER {$$=$1;}

%%
/*Madhumitha S 185001086*/
/*the above gives the productions with precedence in consideration*/

void yyerror(char* s){
    fprintf(stderr,"%s\n",s);
}
void yywrap(){
    return 1;
}
int main(){
    yyparse();//calling the parse function for YACC file
    return 0;
}
```

[redacted]

## 3) OUTPUT SCREENSHOTS:




```
PS C:\Users\seysh\OneDrive\Desktop\madhu_college\6_sem\cd_lab> ./a
3+9
Ans : 12
3+9*6
Ans : 57
(3+4)*7
Ans : 49
(3-4)+(7*6)
Ans : 41
5/7+2
Ans : 2
4^2^1
Ans : 16
(2^3)^2
Ans : 64
4r
invalid character<r>
Ans : 4
0&1
Ans : 0
1&1
Ans : 1
0|1
Ans : 1
1|1
Ans : 1
0?3:4
Ans : 4
1?3:4
```

```
Ans : 0
3==3
Ans : 1
3!=5
Ans : 1
3!=3
Ans : 0
^Z
PS C:\Users\seysh\OneDrive\Desktop\madhu_college\6_sem\cd_lab>
```

**4) LEARNING OUTCOME:**

- Implementing a lex and yacc program for simulating a calculator.
- Coding and implementation in YACC tool.
- Using LEX for token separation and YACC for parsing the tokens to derive meaningful results i.e. here it is the computed values.