**EX.NO.5 DESK CALCULATOR USING YACC**

**Aim:**

To write Lex program to recognize relevant tokens required for the Yacc parser to implement desk calculator.

**cal.l:**

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "y.tab.h"

#include<math.h>

extern int yylval;

%}

%%

[0-9]+ { yylval = atoi(yytext); return INTEGER; }

("+"|"-"|"\*"|"/"|"^"|"("|")"|"\n") { return \*yytext; }

. {

char err[25];

sprintf(err, "Invalid character: %s\n", yytext);

}

**cal.y**

%{

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int yylex(void);

int yyerror(char\* s);

#include "y.tab.h"

%}

%token INTEGER

%%

program: line program

| line

line: expr '\n' { printf("%d\n", $1); }

expr: expr '+' mulex { $$ = $1 + $3; }

| expr '-' mulex { $$ = $1 - $3; }

| mulex { $$ = $1; }

mulex: mulex '\*' powex { $$ = $1 \* $3; }

| mulex '/' powex { $$ = $1 / $3; }

| powex { $$ = $1; }

powex: powex '^' term { $$ = pow($1, $3); }

| term { $$ = $1; }

term: '(' expr ')' { $$ = $2; }

| INTEGER { $$ = $1; }

;

%%

int yyerror(char\* s)

{

fprintf(stderr, "%s\n", s);

return 0;

}

int yywrap()

{

return 1;

}

int main()

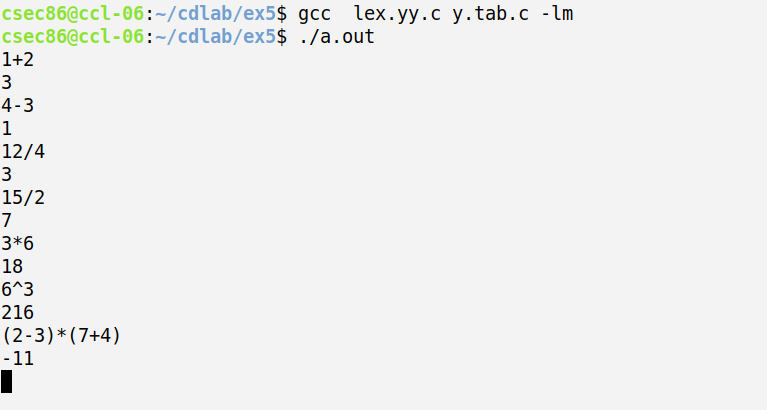
{

yyparse();

return 0;

}

**OUTPUT:**

****