/**

EXPERIEMENT NO : 08 (GROUP – B)
TITLE : Write a program using YACC specifications to implement syntax analysis phase of compiler to validate type and syntax of variable declaration in Java

NAME : PROF. ANAND N. GHARU

ROLL NO :

CLASS : TE COMPUTER **DATE** : /01/2018

**********/

Lex Progarm

```
%{
#include"y.tab.h"
#include<math.h>
extern yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext);return NUM;}
[+] {return '+';}
[-] {return '-';}
[*] {return '*';}
[/] {return '/';}
[\t]+;
[\n] {return 0;}
. {return yytext[0];
}
%%
```

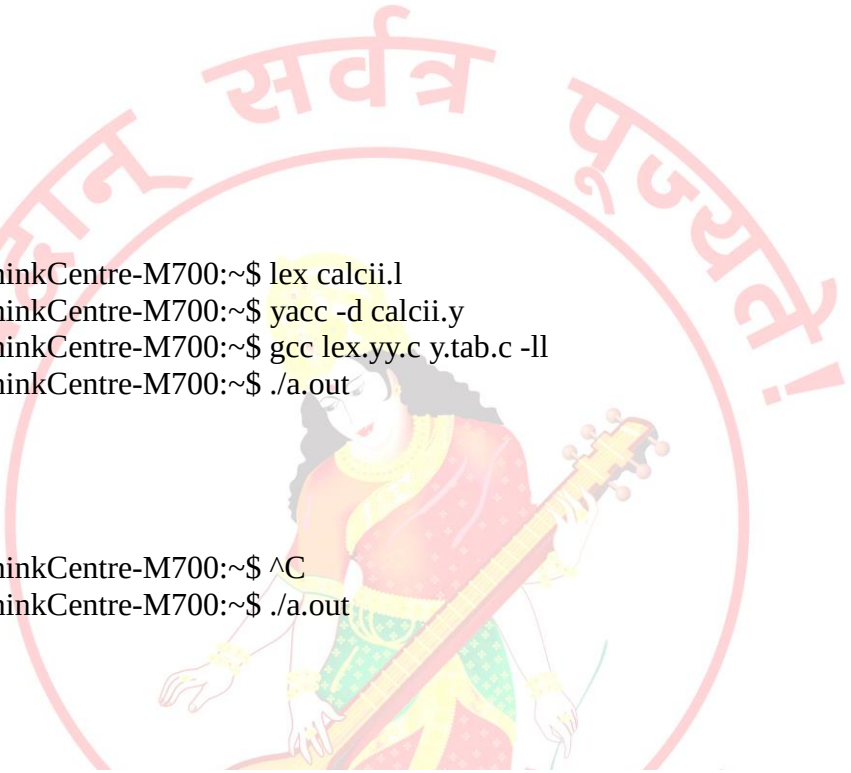
Yacc Program

```
%{
#include<stdio.h>
%}
%token NUM
%left '-''+'
%right '*''/'
%%
start: exp {printf("%d\n",$$);}
exp:exp'+'exp {$$=$1+$3;}
|exp'-'exp {$$=$1-$3;}
|exp'*'exp {$$=$1*$3;}
|exp'/'exp
{
if($3==0)
yyerror("error");
else
{
$$=$1/$3;
}
}
|'('exp')' {$$=$2;}
```

```
|NUM {$$=$1;}
;
%%
main()
{
printf("Enter the Expression\n");
if(yyparse()==0)
printf("Correct answer\n");
}
yywrap(){}
yyerror()
{
printf("Invalid Input\n");
}
```

//Output

```
pvgcoen-6@pvgcoen6-ThinkCentre-M700:~$ lex calcii.l
pvgcoen-6@pvgcoen6-ThinkCentre-M700:~$ yacc -d calcii.y
pvgcoen-6@pvgcoen6-ThinkCentre-M700:~$ gcc lex.yy.c y.tab.c -ll
pvgcoen-6@pvgcoen6-ThinkCentre-M700:~$ ./a.out
Enter the Expression
1.5+1.6
1
Invalid Input
pvgcoen-6@pvgcoen6-ThinkCentre-M700:~$ ^C
pvgcoen-6@pvgcoen6-ThinkCentre-M700:~$ ./a.out
Enter the Expression
8+5*11
63
Correct answer
```



// infix.l

```
%{
#include "y.tab.h"
%}
%%
[0-9]+  { yylval.dval=atoi(yytext);  return NUMBER;}
[0-9]*"."[0-9]+  { yylval.dval=atof(yytext);  return NUMBER;}
[a-zA-Z] { return LETTER; }
"+" { return PLUS;}
"-" { return MINUS;}
"*" { return MULTIPLY;}
"/" { return DIVIDE;}
"(" { return OPEN;}
")" { return CLOSE;}
"\n" { return ENTER;}
"$" { return 0;}
%%
```

//infix.y

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

%union {
double dval;
char symbol;
}
%token<dval>NUMBER
%token<symbol>LETTER
%token PLUS MINUS MULTIPLY DIVIDE OPEN CLOSE ENTER
%left PLUS MINUS
%left DIVIDE MULTIPLY
%nonassoc UMINUS
%type<dval>E

%%
print: E ENTER { printf("\n\v VALID INFIX EXP.....\n"); exit (0); }
;
E:E PLUS E
|
E MINUS E
|
E MULTIPLY E
|
E DIVIDE E
|
MINUS E %prec UMINUS {$$=-$2;}
|
OPEN E CLOSE { $$=$2;}
|
NUMBER { $$=$1; }
|
LETTER {$$=$1;}
;
%%

void yyerror( char *msg)
{
  printf("\n INVALID INFIX EXPRESSION.....:  ");
```

```
}
int main()
{
 printf("\n Enter infix expression:  ");
  yyparse();
  return 0;
}
```

//OUTPUT

```
/*****************Output**********************/
[pvg@localhost ~]$ vi infix.l
[pvg@localhost ~]$ vi infix.y
[pvg@localhost ~]$ lex infix.l
[pvg@localhost ~]$ yacc -d infix.y
[pvg@localhost ~]$ cc lex.yy.c y.tab.c -ll
[pvg@localhost ~]$ ./a.out

 Enter infix expression:  a+b*c-d


 VALID INFIX EXP.......
[pvg@localhost ~]$ lex infix.l
[pvg@localhost ~]$ yacc -d infix.y
[pvg@localhost ~]$ cc lex.yy.c y.tab.c -ll
[pvg@localhost ~]$ ./a.out

 Enter infix expression:  -+*a+b-c*d

 INVALID INFIX EXPRESSION....
```