Exp 8 : CALCULATOR USING YACC

LEX

1. Start
2. [0-9]+ :Match sequences of digits , convert the matched string to an integer (atoi(yytext)) and store it in yylval , return the token type NUM
3. [\t\n] :Match tabs and newlines and return 0, indicating that they should be ignored.
4. . :Match any character that hasn't been matched by the previous rules. Return the matched character itself.

5. int yywrap() - return 1, indicating the end of input
6. Stop

YACC

1. Start
2. %token NUM
3. %left '+' '-' '*' '/' '%' '(' ')'
   Define the precedence and associativity of operators.

4. S -> E
   E -> E + E { $$ = $1 + $3 } | E-E { $$ = $1 - $3 } | E*E { $$ = $1 * $3 } | E/E {
   if ($3 != 0) { $$ = $1 / $3 } else { yyerror() } | E%E { if ($3 != 0) { $$ = $1 % $3} else { yyerror() }} | (E) { $$ = $2 } | NUM { $$ = $1 }

5. int yyerror() - Error handling function called on parsing error
6. In main
   - Print a prompt asking the user to enter an arithmetic expression
   - Call yyparse() to start the parsing proces
7. Stop

Calculator (Lex)

```
%{
   #include<stdio.h>
   #include "y.tab.h"
   extern int yylval;
%}

%%
[0-9]+ { yylval = atoi(yytext); return NUM; }
[\t\n] {;}
. { return yytext[0]; }
%%

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

Calculator (YACC)

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

%token NUM
%left '+' '-'
%left '*' '/' '%'
%left '(' ')'

%%
S: E {
   printf("Result = %d\n", $$);
   return 0;
};

E: E '+' E { $$ = $1 + $3; }
| E'-'E { $$ = $1 - $3; }
| E'*'E { $$ = $1 * $3; }
| E'/'E {
   if($3 != 0){
      $$ = $1 / $3;
   }else{
      yyerror();
   }
}
| E'%'E {
   if($3 != 0){
      $$ = $1 % $3;
   }else{
      yyerror();
   }
}
| '(' E ')' { $$ = $2; }
| NUM { $$ = $1; }
;
%%

int yyerror(){
   printf("Entered arithmetic expression is Invalid\n");
   exit(0);
}

int main(){
   printf("Enter the Arithmetic Expression : ");
   yyparse();
   return 0;
}
```

output

```
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/2 Calculator$ flex calculator.l
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/2 Calculator$ yacc -d calculator.y
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/2 Calculator$ gcc lex.yy.c y.tab.c -o calculator
y.tab.c: In function ‘yyparse’:
y.tab.c:1024:16: warning: implicit declaration of function ‘yylex’ [-Wimplicit-function-declaration]
 1024 |       yychar = yylex ();
      |                ^~~~~
calculator.y:24:9: warning: implicit declaration of function ‘yyerror’; did you mean ‘yyerrok’? [-Wimplicit-function-declar
ation]
   24 |         yyerror();
      |         ^~~~~~
      |         yyerrok
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/2 Calculator$ ./calculator
Enter the Arithmetic Expression : 6/2+4-3*(2+1)
Result = -2
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/2 Calculator$ ./calculator
Enter the Arithmetic Expression : 3+4*/5
Entered arithmetic expression is Invalid
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/2 Calculator$ ./calculator
Enter the Arithmetic Expression : 1+2*3/0
Entered arithmetic expression is Invalid
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/2 Calculator$
```