CODE

calculator.y

```
%{
   void yyerror();
%}

%token INTEGER
%left '+' '-'
%left '*' '/'

%%

program:
   program statement '\n'
   |
   ;

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

expr: INTEGER
   | expr '+' expr { $$ = $1 + $3; }
   | expr '-' expr { $$ = $1 - $3; }
   | expr '*' expr { $$ = $1 * $3; }
   | expr '/' expr { $$ = $1 / $3; }
   | '(' expr ')' { $$ = $2; }
   ;
%%

void yyerror(){

}
int main(void){
   yyparse();
   return 0;
}
```

calculator.l

```
%{
   #include<stdio.h>
   #include<stdlib.h>
   #include "y.tab.h"
```

```
   int yyerror();
   extern int yylval;
%}

%%

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

[-+()=/*\n] {return *yytext;}
[ \t]   ;

. yyerror();
%%

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

OUTPUT

```
➜  Lab 8 git:(master) flex calculator.l
➜  Lab 8 git:(master) yacc -d calculator.y
➜  Lab 8 git:(master) gcc lex.yy.c y.tab.c
y.tab.c: In function ‘yyparse’:
y.tab.c:1220:16: warning: implicit declaration of function ‘yylex’ [-Wimplicit-function-declaration]
 1220 |       yychar = yylex ();
      |                ^~~~~
calculator.y:16:18: warning: implicit declaration of function ‘printf’
[-Wimplicit-function-declaration]
   16 | statement: expr {printf("%d\n",$1);}
      |                  ^~~~~~
calculator.y:16:18: warning: incompatible implicit declaration of built-in function ‘printf’
calculator.y:16:1: note: include ‘<stdio.h>’ or provide a declaration of ‘printf’
   15 |
  +++ |+#include <stdio.h>
   16 | statement: expr {printf("%d\n",$1);}


➜  Lab 8 git:(master) ./a.out
```

```
2+3
5
2-3
-1
2+4
6
2+4*2
10
(2+4)
6
(2+8)*2
20
^C
```