

## Calc.y

```
% {  
#include <stdlib.h>  
#include <stdio.h>  
int yylex(void);  
#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 '*' term { $$ = $1 * $3; }
```

```
  | mulex '/' term { $$ = $1 / $3; }
```

```
  | term { $$ = $1; }
```

```
term:
```

```
  '(' expr ')' { $$ = $2; }
```

```
  | INTEGER { $$ = $1; }
```

```
%%
void yyerror(char *s)
{
    fprintf(stderr,"%s\n",s);
    return;
}
yywrap()
{
    return(1);
}
```

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

## **Lexx.l**

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

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

```
[-* /] {return *yytext;}  
"(" {return *yytext;}  
")" {return *yytext;}  
\n {return *yytext;}  
. {char msg[25];  
  sprintf(msg,"%s <%s>","invalid character",yytext);  
  yyerror(msg);}
```

### **Output :-**

```
lex lexx.l  
yacc -d calc.y  
gcc y.tab.c lex.yy.c  
./a.out
```