

# Lex and Yacc for simple Calculator

## Lex:

```
/ Lex rules for simple
calculator %{
/ including required libraries and
files #include<stdlib.h>
#include
"y.tab.h" extern
int yylval; %}

/      representations of the symbols
%%

[0-9]+ {yylval=atoi(yytext); return
NUMBER;} ">=" return GE ;
"<=" return LE ;
"!=" return NE ;
"==" return
EQ ; [\n] return
0 ; [\t];
. return yytext[0];
%%
```

## Yacc:

/ Yacc grammar for simple

calculator %{

#include<stdio.h>

%}

%token NAME

NUMBER %left GE LE NE

EQ '<' '>' '%' %left '-' '+'

%left '\*'

'/' %nonassoc

UMINUS

%%

statement:NAME '=' exp

|exp {printf("=%d\n",\$1);}

;

exp:NUMBER { \$\$ = \$1;}

|exp '+' exp { \$\$ = \$1 + \$3 ;}

|exp '-' exp { \$\$ = \$1 - \$3 ;}

|exp '\*' exp { \$\$ = \$1 \* \$3 ;}

|exp '/' exp { \$\$ = \$1 / \$3 ;}

|exp '<' exp { \$\$ = \$1 < \$3 ;}

|exp '>' exp { \$\$ = \$1 > \$3 ;}

|exp '%' exp { \$\$ = \$1 % \$3 ;}

|exp GE exp { \$\$ = \$1 >= \$3 ;}

|exp LE exp { \$\$ = \$1 <= \$3 ;}

|exp NE exp { \$\$ = \$1 != \$3 ;}

|exp EQ exp { \$\$ = \$1 == \$3 ;}

| '-' exp %prec UMINUS { \$\$ = -

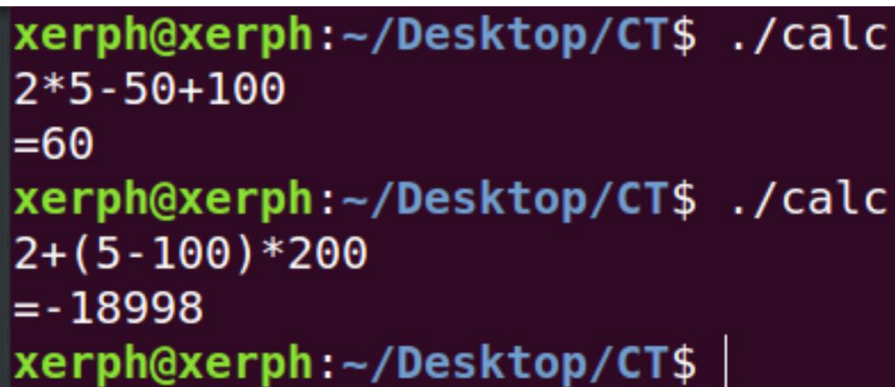
\$2 ;} | '(' exp ')' { \$\$ = \$2;}

%%

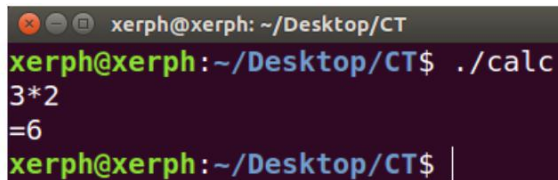
int main()

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

## OUTPUT:

A terminal window with a dark purple background. The prompt is 'xerph@xerph: ~/Desktop/CT\$'. The user enters './calc', followed by '2\*5-50+100', and the output is '=60'. Then the user enters '2+(5-100)\*200', and the output is '=-18998'. The prompt is followed by a vertical bar character '|'.

```
xerph@xerph:~/Desktop/CT$ ./calc
2*5-50+100
=60
xerph@xerph:~/Desktop/CT$ ./calc
2+(5-100)*200
=-18998
xerph@xerph:~/Desktop/CT$ |
```

A terminal window with a dark purple background. The title bar shows 'xerph@xerph: ~/Desktop/CT'. The prompt is 'xerph@xerph: ~/Desktop/CT\$'. The user enters './calc', followed by '3\*2', and the output is '=6'. The prompt is followed by a vertical bar character '|'.

```
xerph@xerph:~/Desktop/CT$ ./calc
3*2
=6
xerph@xerph:~/Desktop/CT$ |
```