Calc.l

%{

#include "y.tab.h"

#include <stdio.h>

#include <stdlib.h>

extern int yylval;

%}

%%

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

['+'|"\-"|'\*'|'/'|'^'|'%'|'('|')'|'\n'|'<'|'>'|'!'|'&'|'|'] { return (\*yytext);}

"<=" return A;

">=" return B;

"==" return C;

"!=" return D;

"&&" return F;

"||" return G;

"<<" return H;

">>" return I;

"\t" ;

. {printf("invalid char");}

%%

int yywrap()

{

return 1;

}

Calc.y

%{

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int yylex(void);

void yyerror();

%}

%token NUM A B C D F G H I

%left G

%left F

%left '|'

%left '&'

%left C D

%left '>' '<' A B

%left H I

%left '+' '-'

%left '\*' '/' '%'

%right '^'

%right '!'

%left '(' ')'

%%

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

E : E '+' E { $$ = $1 + $3; }

| E '-' E { $$ = $1 - $3; }

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

| E'/'E { $$ = $1 / $3; }

| E'%'E { $$ = $1 % $3; }

| E'^'E { $$ = pow($1,$3); }

|'('E')' { $$ = $2; }

| '!' E { $$ = (!($2)); }

| E'<'E {$$ = $1 < $3; }

| E'>'E {$$ = $1 > $3; }

| E A E {$$ = $1 <= $3;}

| E B E {$$ = $1 >= $3;}

| E C E {$$ = $1 == $3;}

| E D E {$$ = $1 != $3;}

| E F E {$$ = $1 && $3;}

| E G E {$$ = $1 || $3;}

| E '&' E { $$ = $1 & $3; }

| E '|' E { $$ = $1 | $3; }

| E H E { $$ = $1 << $3; }

| E I E { $$ = $1 >> $3; }

| '-' E { $$ = (-($2));}

| NUM { $$ = $1; }

;

%%

void yyerror()

{

printf("\nInvalid exp\n");

}

int main () {

//yyin = fopen("calcip","r");

printf("EXPRESSION : ");

yyparse ();

return 0;

}