

```

%{
    #include "assignment.tab.h"
    #include <stdio.h>
    #include <string.h>
    #include <stdlib.h>
    #include <math.h>
}%

header          INC[ ]+[<][^\n]+[.H][>]
keyword          NUM|String|Loop|IF|Else|to|FOR
operator         pls|mns|mul|div|gt|lt|mod|eq|sin|cos|log|tan
punctuation      lb|rb|lp|rp|cm|sm
v                [a-z]
identifier        [a-zA-Z_]+[0-9]*
num               [+~]?[0-9]+
DOUBLE            [0-9]*[.][0-9]*
function          []['][a-zA-Z_]+[0-9]*[']

%%
{header}        {}
{keyword}        {if(strcmp(yytext,"NUM")==0)  return NUMBER;
                  else
if(strcmp(yytext,"String")==0)  return STRING;
                  else if(strcmp(yytext,"IF")==0)
return IF;
                  else if(strcmp(yytext,"Else")==0)
return ELSE;
                  else if(strcmp(yytext,"Loop")==0)
return LOOP;
                  else if(strcmp(yytext,"to")==0)
return TO;
                  else if(strcmp(yytext,"FOR")==0)
return FOR;
                  }
{operator}       {if(strcmp(yytext,"pls")==0)  return PLUS;
                  else if(strcmp(yytext,"mns")==0)
return MINUS;
                  else if(strcmp(yytext,"div")==0)
return DIVISION;
                  else if(strcmp(yytext,"mul")==0)
return MULTIPLICATION;
                  else if(strcmp(yytext,"gt")==0)
return GT;
                  else if(strcmp(yytext,"lt")==0)
return LT;
                  else if(strcmp(yytext,"eq")==0)
return EQUAL;
                  else if(strcmp(yytext,"mod")==0)
return MOD;

```

```

return SIN;
return COS;
return TAN;
return LOG;}
{punctuation}
return SM;
return LP;
return RP;
return LB;
return RB;}
"\n"
"("
")"
{"
"}"
{num}
{DOUBLE}
{v}
{identifier}
{function}
.
%%
int yywrap()
{
    return 1;
}

else if(strcmp(yytext,"sin")==0)
else if(strcmp(yytext,"cos")==0)
else if(strcmp(yytext,"tan")==0)
else if(strcmp(yytext,"log")==0)
{if(strcmp(yytext,"cm")==0) return CM;
else if(strcmp(yytext,"sm")==0)
else if(strcmp(yytext,"lp")==0)
else if(strcmp(yytext,"rp")==0)
else if(strcmp(yytext,"lb")==0)
else if(strcmp(yytext,"rb")==0)
{return NEWLINE;}
{return LPP;}
{return RPP;}
{return LBP;}
{return RBP;}
{
    sscanf(yytext, "%d", &yylval);
    return NUMB;
}
{
    sscanf(yytext, "%lf", &yylval);
    return DOUBLE;
}
{
    yylval = *yytext - 'a';
    return IDENTIFIER1;
}
{return IDENTIFIER;}
{}

```