```
%{

typedef struct
{
 char q[41624];
 char thestr[25];
 int ival;
 int ttype;
}tstruct ;

#define YYSTYPE  tstruct

#include "w.tab.h"
#define YY_USER_ACTION yylloc.first_line = yylloc.last_line =
yylineno;
%}

%option yylineno
%option caseless

DIGIT  [0-9]
UC     [A-Z]
LC     [a-z]
L      [A-Za-z]

%%

ALPHA     { return tstart;}
OMEGA     { return tfinish;}
MAXX       { return tmaxx;}
MAXY       { return tmaxy;}
MINX       { return tminx;}
MINY       { return tminy;}
INT        { return tint;}
FLOAT      { return tfloat;}
PUT        { return tput;}
GET        { return tget;}
GRAPH      { return tgraph;}
SOLVELIN  { return tsollin;}
SOLVEQUAD { return tsolquad;}
SOLVECUBE { return tsolcube;}
FOR        { return tfor;}
WHILE      { return twhile;}
IF         { return tif;}
ELSE       { return telse;}
\{         { return tob;}
\}         { return tcb;}

\"([^"]*)\"  { strcpy( yylval.thestr, yytext);  return tstrlit; }
```

```
{L}+     { strcpy( yylval.thestr, yytext); return tid; }
-?{DIGIT}+ { strcpy( yylval.thestr, yytext); return tnum; }
{DIGIT}*\.{DIGIT}+    {strcpy( yylval.thestr, yytext); return tfnum;}
(<|>|(==)|(<=)|(>=)|(!=)) {strcpy( yylval.thestr, yytext); return
tbool;}

@@.*\n   {} /* comments  */


[ \t]     /* ignore whitespace */

\n+       {return tsemi;}


<<EOF>>   yyterminate();  /* signal end of dialogue */

.         return yytext[0];

%%


void yyerror(char *s)  /* Called by yyparse on error */
{
  printf ("\terror: %s\n", s);
  printf ("ERROR: %s at line %d\n", s, yylineno);
}
```