```
%{

/*
 * ===============================================================================
 *
 * check2.l
 *
 * Programmer --- Patrick Fischer
 * Assignment --- Checkpoint 2
 * Due        --- 9/22/16
 *
 * ===============================================================================
 */

#include <stdio.h>

%}

%%

[\t ]+                                  { }

(m|M)(a|A)(i|I)(n|N)                    { printf("RWMAIN: %s\n", yytext); }

(e|E)(x|X)(i|I)(t|T)                    { printf("RWEXIT: %s\n", yytext); }

(i|I)(f|F)                              { printf("RWIF: %s\n", yytext); }

(e|E)(l|L)(s|S)(e|E)                    { printf("RWELSE: %s\n", yytext); }
```

```
(w|W)(h|H)(i|I)(l|L)(e|E)                           { printf("RWWHILE: %s\n", yytext); }
(c|C)(o|O)(u|U)(n|N)(t|T)(i|I)(n|N)(g|G)            { printf("RWCOUNTING: %s\n", yytext); }
(e|E)(n|N)(d|D)                                     { printf("RWEND: %s\n", yytext); }
(d|D)(a|A)(t|T)(a|A)                                { printf("RWDATA: %s\n", yytext); }
(a|A)(l|L)(g|G)(o|O)(r|R)(i|I)(t|T)(h|H)(m|M)       { printf("RWALG: %s\n", yytext); }
[-+]?[0-9]+                                         { printf("LITINT: %s\n", yytext); }
[-+]?([0-9]*\.?[0-9]+|[0-9]+\.)((e|E)[-+]?[0-9]+)? { printf("LITREAL: %s\n", yytext); }
(i|I)(n|N)(t|T)(e|E)(g|G)(e|E)(r|R)                 { printf("RWINT: %s\n", yytext); }
(r|R)(e|E)(a|A)(l|L)                                { printf("RWREAL: %s\n", yytext); }
(l|L)(i|I)(s|S)(t|T)                                { printf("RWLIST: %s\n", yytext); }
(u|U)(p|P)(w|W)(a|A)(r|R)(d|D)                      { printf("RWUPWARD: %s\n", yytext); }
(d|D)(o|O)(w|W)(n|N)(w|W)(a|A)(r|R)(d|D)            { printf("RWDOWNWARD: %s\n", yytext); }
(t|T)(o|O)                                          { printf("RWTO: %s\n", yytext); }
(r|R)(e|E)(a|A)(d|D)                                { printf("RWREAD: %s\n", yytext); }
(p|P)(r|R)(i|I)(n|N)(t|T)                           { printf("RWPRINT: %s\n", yytext); }
```

```
\"([^\\\"]|\\.)*\"          { printf("CHARSTRING: %s\n", yytext); }
#.*                         { printf("COMMENT: %s\n", yytext); }
!                           { printf("CARRETURN: %s\n", yytext); }
":="                        { printf("ASSIGN: %s\n", yytext); }
,                           { printf("COMMA: %s\n", yytext); }
";"                         { printf("SEMICOLON: %s\n", yytext); }
":"                         { printf("COLON: %s\n", yytext); }
"("                         { printf("LPAREN: %s\n", yytext); }
")"                         { printf("RPAREN: %s\n", yytext); }
"["                         { printf("LBRACK: %s\n", yytext); }
"]"                         { printf("RBRACK: %s\n", yytext); }
"+"                         { printf("ADD: %s\n", yytext); }
"-"                         { printf("MINUS: %s\n", yytext); }
"*"                         { printf("MULT: %s\n", yytext); }
"/"                         { printf("DIV: %s\n", yytext); }
```

```
"<"                                 { printf("LESS: %s\n", yytext); }

">"                                 { printf("GREATER: %s\n", yytext); }

"<="                                { printf("LESSEQU: %s\n", yytext); }

">+"                                { printf("GREATEQU: %s\n", yytext); }

"="                                 { printf("EQUAL: %s\n", yytext); }

"<>"                                { printf("NOTEQUAL: %s\n", yytext); }

"&"                                 { printf("AND: %s\n", yytext); }

"|"                                 { printf("OR: %s\n", yytext); }

"~"                                 { printf("NOT: %s\n", yytext); }

([a-zA-Z][a-zA-Z0-9]*)+             { printf("VAR: %s\n", yytext); }

.|\n                                { }

%%

int main()
{
   yylex();
   return 0;
}
```