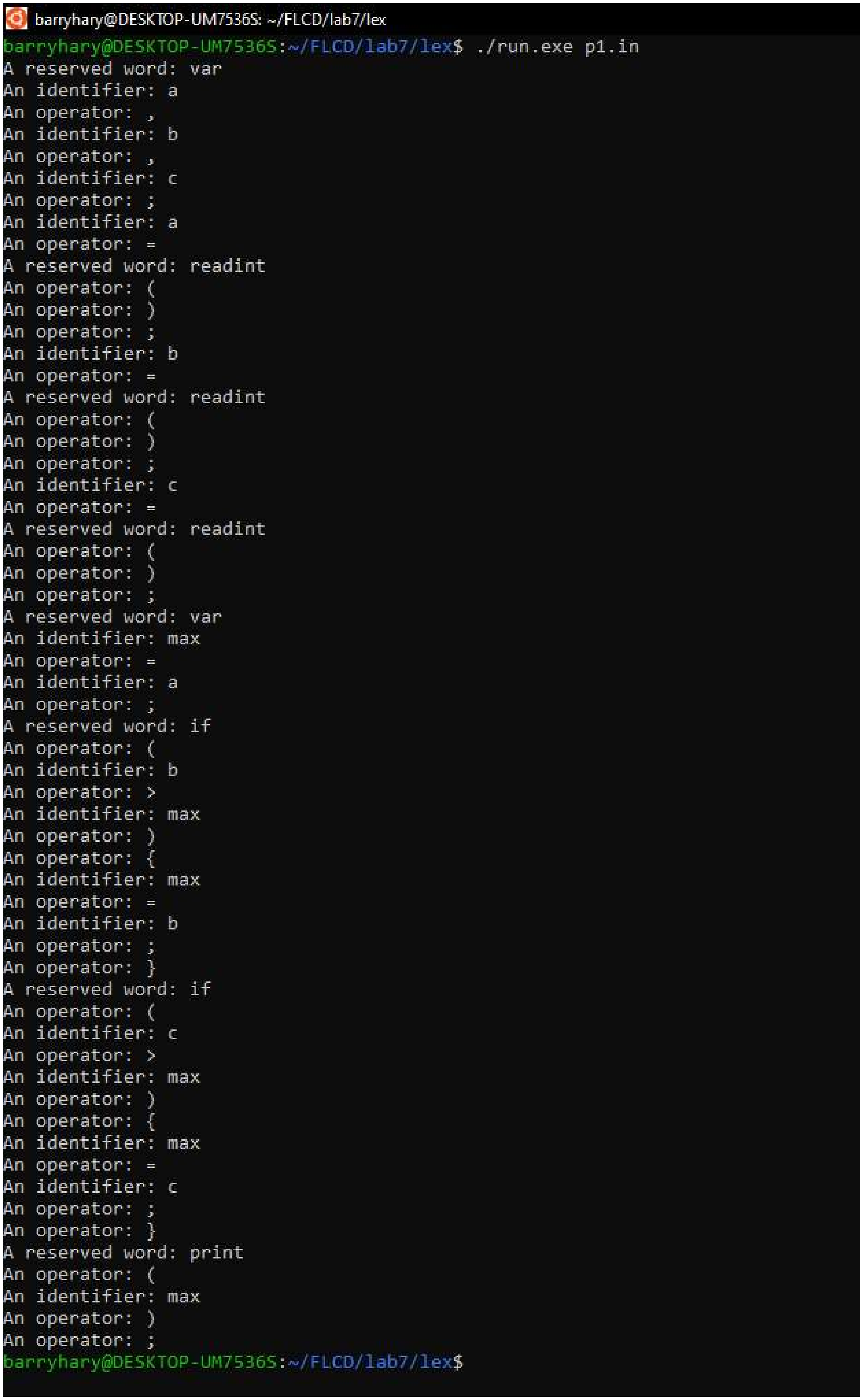
LEX

```
%{
#include <math.h>
%}

%option noyywrap

DIGIT [0-9]
ID [a-z][a-z0-9]*

%%

{DIGIT}+ printf("An integer: %s (%d)\n", yytext, atoi( yytext ) );

{DIGIT}+"."{DIGIT}*  printf( "A float: %s (%g)\n", yytext, atof( yytext ) );

"if"|"readint"|"var"|"list"|"append"|"while"|"print"  printf( "A reserved word: %s\n", yytext );

{ID} printf( "An identifier: %s\n", yytext );

"+"|"-"|"*"|"/"|","|"="|"("|")"|"{"|"}"|";"|"<"|">"|"<="|">="|"=="|"!="|"."|"["|"]" printf( "An
operator: %s\n", yytext );

"{"[^}\n]*"}" /* eat up one-line comments */

[ \t\n]+ /* eat up whitespace */

. printf("Lexical Error: %s\n",yytext);
%%
int main(int argc, char **argv )
{
    ++argv, --argc; /* skip over program name */
    if ( argc > 0 )
    yyin = fopen( argv[0], "r" );
    else
     yyin = stdin;
    yylex();
}
```



```
barryhary@DESKTOP-UM7536S:~/FLCD/lab7/lex$ ./run.exe p1.in
A reserved word: var
An identifier: a
An operator: ,
An identifier: b
An operator: ,
An identifier: c
An operator: ;
An identifier: a
An operator: =
A reserved word: readint
An operator: (
An operator: )
An operator: ;
An identifier: b
An operator: =
A reserved word: readint
An operator: (
An operator: )
An operator: ;
An identifier: c
An operator: =
A reserved word: readint
An operator: (
An operator: )
An operator: ;
A reserved word: var
An identifier: max
An operator: =
An identifier: a
An operator: ;
A reserved word: if
An operator: (
An identifier: b
An operator: >
An identifier: max
An operator: )
An operator: {
An identifier: max
An operator: =
An identifier: b
An operator: ;
An operator: }
A reserved word: if
An operator: (
An identifier: c
An operator: >
An identifier: max
An operator: )
An operator: {
An identifier: max
An operator: =
An identifier: c
An operator: ;
An operator: }
A reserved word: print
An operator: (
An identifier: max
An operator: )
An operator: ;
barryhary@DESKTOP-UM7536S:~/FLCD/lab7/lex$
```