Christian McGovern, The problem for this lab was printing out the error and the line number on the .tst file where the error occurred.

I made about 3 lines of code changes to the original program, First I created a int to count the lines, then I created a new rule to increment the int whenever there is a new line: \n {printf("\n",linenum++);}

I also had to print a \n in order to refresh the buffer.

I added the linenum to the error printf() and printed out the line number for the error.

Code:

/\* simple lex program which removes comments from a source program

The main key is that a variable "comment" is set when the start of a comment

is seen and then unset when the ending set is seen. It is possible to have

two starts closed by on end.

Shaun Cooper

January 2015

\*/

int comment = 0;

int debug=0; /\* prints out debug statements if desired \*/

/\*

\* Christian McGovern

\* Lab2.1 1/22/18

\*

\*/

int linenum = 1;//new variable for counting lines

%%

\n {printf("\n",linenum++);} //rule to increment linenum on each line

"/\*" {

if (comment) fprintf(stderr,

">>>>>>>> line %d: Possible Nested comment <<<<<<<<<<\n", linenum);//printing linenum and removed debug

comment = 1;

}

"\*/" {

if (!comment) printf("%s",yytext); /\*print out if it is not in a comment \*/

comment = 0;

}

. { if (!comment) printf("%s",yytext);}

%%

int yywrap(void)

{

return 1;

}

main()

{

yylex();

}

