Compiler Construction Labsheet-5

yymore(): Appends the next matched string to the current value of the yytext array
rather than replacing the contents of the yytext array.
Ex:

```
%%
aECHO;yymore();
bECHO;
%%
```

Input: ab

Observe output.

Ex:

```
%%
bits- {ECHO; yymore();}
hyd ECHO;
```

If bits-hyd is input what does flex print?

yyless(): trim characters from yytext and puts them back on stdin.

yyless(n) returns all but the first n characters of the current token back to the input stream, where they will be rescanned when the scanner looks for the next match. yytext and yyleng are adjusted appropriately (e.g., yyleng will now be equal to n).
For example, on the input "foobar" the following will write out "foobarbar":

```
%%
foobar {ECHO; yyless(3);}
[a-z]+ ECHO;
%%
```

Input foobar and observe the output and try to reason out.

Matching a Right Context

Lex can match a right context while matching a regexp by using the notation r1/r2 where the regexp r1 will match only if the right context matches the regexp r2 . The longest match is computed using r1 and r2 concatenated together. The lexeme reported is the match for r1 .

For example, consider the following lex program.

```
%%

a+/b  { printf("T_AB: %s\n", yytext); }

a+/c  { printf("T_AC: %s\n", yytext); }

b     { printf("T_B: %s\n", yytext); }

c     { printf("T_C: %s\n", yytext); }

.  ECHO;

\n return 1;

%%

int main()

{

yylex();

}
```

On input: ab
output: T_AB:a
T_B:b
On input: ac
output: T_AC:a
T_C:c
On input: abc
output: T_AB:a
T_B:b
T_C:c

More Excercise problems:

1. Write a lex program to count and remove comments from a C file (write the output to a new C file). (Assume that there is no "*/" in between "/*" and "*/").

2. Find the number of valid 'printf' statements alongwith variables used in it

3. Lex program to implement a simple calculator

4. Lex Program that Checks a vaild arithmentc expression