1. Write BNF grammar for the **for** loop statement of **C**. Assume this is the 89 standard that does NOT allow new variables as C++ did (not allowed is the "**int"** in **for (int i=0; i<5; i++)**. Also assume block is NOT the same as a statement (different books take different take on this) - block is {} with variables and statements inside, at least one statement assume. Further assume <expression> and <statement> and <block> nonterminals are already defined (you dont' define them, just use them).  Do not use the extended BNF notation.
2. Design a regular language where every sentence has to start with any number of strings 101 (any number is none or more), then repeats 00 any number of times, then repeats 01 at least once. Use regular expression notation.  
   Clarification: alphabet is {0,1}. 'string' is the same as program in this context, but here the programs are written using binary alphabet so strings here are binary.
3. For the above, how many valid strings there are?
4. For the above, how many valid strings are there if we restrict the total length to a) at most 7 bits?

[Previous](https://umsl.instructure.com/courses/64310/modules/items/1043236)[Next](https://umsl.instructure.com/courses/64310/modules/items/1124811)