TCSS 333 HW1 Summer 2017

Due: by midnight, Thurs. June 29 Turn in your .c file to Canvas.
(or with 10% late penalty, by midnight, Fri. June 30)

For this assignment, you will write an interactive program that codes and decodes messages. The program should respond to 3 commands: c/C for code, d/D for decode, and q/Q for quit.
In the case of code and decode:

- the program reads in a message (stopping at end of line)
- code or decodes that message and
- prints out the result (the coded or decoded version of the original message)

The encoding replaces every input character with 7 new ones. The original character is assumed to have an ASCII value less than 128, therefore it can be thought of as a 7 bit binary number. This binary number is then disguised as 7 seemingly random upper case letters. However the 7 bits are actually encoded in those letters: letters with an even ASCII value represent 0's, letters with an odd ASCII value represent 1's. Thus the letter 'W' which has an ASCII value of 87 (or in binary 101 0111) could be encoded by the letters SLIFEMY because L and F have even ASCII values and the other letters have odd values. Decoding reverses this process, therefore a coded message has only capital letters and the number of letters is always an even multiple of 7. Your encodings should look random and for that you must use the `rand()` function. Your program can assume all inputs are legal. Here is an example of the program running:

```
[jhmayer@cssgate2 ~]$ ./hw1X
> c
? TCSS 333
WLMBCDROHDDDMOWFSJBESARCBNECDYXXPLRLMAPFWKHOKHNWKHSQBBUQ
> c
? TCSS 333
CLIVSDTIXVTRSWQFMFDWSOFSBNUQHSXPHHZVKMLNOKPQKTZYABKINDOM
> c
? TCSS 333
GDWDWFPIVTDLCGEHAHRQKWPQBNYQXGDPBTRBSADDUETOKXHOAHWMXXYM
> d
? WLMBCDROHDDDMOWFSJBESARCBNECDYXXPLRLMAPFWKHOKHNWKHSQBBUQ
TCSS 333
> d
? CLIVSDTIXVTRSWQFMFDWSOFSBNUQHSXPHHZVKMLNOKPQKTZYABKINDOM
TCSS 333
> d
? GDWDWFPIVTDLCGEHAHRQKWPQBNYQXGDPBTRBSADDUETOKXHOAHWMXXYM
TCSS 333
> q
[jhmayer@cssgate2 ~]$
```

This problem can be solved in less than 100 lines of code. No data structures are needed, not even arrays. I encourage you to process the input and output one character at a time.

Basic Code Quality Requirements (these apply to future assignments as well)

Code you find on the internet cannot be used in any assignment.

Your code must include your name in a comment at the top of the program!

Write multiple functions that each do one simply described thing. If it's not possible to see all the code of a function at once (without scrolling), the function is probably too long.

C requires you to define a function before you call it. If function A calls another function B, B should appear above A. (main naturally ends up at the bottom.) Or add prototypes to the top of your .c file if you wish.

The names you choose for variables, functions, etc. should be meaningful and descriptive.

Redundant code is code that is clearly repetitive and doesn't need to be. If you find yourself copying and pasting while coding, you very likely are writing some redundant code. Redundant code should be cleaned up before submission. Use functions and loops to eliminate redundancy.

Your code must be neatly and correctly indented. Don't use the tab key to indent. Tab doesn't have the same effect in every editor, so code that is beautifully indented in one editor can look bad in another. The text book is a good model for indenting.

Converting a decimal value to a 7 bit binary number:

```
Start with a decimal value d and pow2 set to 64  (2 ^ 7).
Repeat 7 times:
    If d is less than pow2, print out a 0.
    Otherwise print out a 1 and reduce d by pow2.
    In any case, divide pow2 by 2 before continuing.
```

Example: the decimal value is 87

d	pow2	print	new d
87	64	1	23
23	32	0	23
23	16	1	7
7	8	0	7
7	4	1	3
3	2	1	1
1	1	1	0

Creating a decimal value from a 7 bit binary number:

```
Start with v = 0 and pow2 set to 64.
Repeat 7 times:
    If the next bit is 1, add pow2 to v.
    Divide pow2 by 2
```

Example: The seven bits are 1010111

v	pow2	next bit	new v
0	64	1	64
64	32	0	64
64	16	1	80
80	8	0	80
80	4	1	84
84	2	1	86
86	1	1	87
87			