**CS-251 - Week 05**

## Some fun with recursion!

Consider all the bit strings of a certain length. For example length 3:

000

001

010

011

100

101

110

111

The above listing of all 8 such strings is in a particular order -- the numerical order if the strings are interpreted as base-2 integers.

Of course we can order them any way we want.

Suppose we want to order them such that each string ***differs from the string preceding it in exactly one bit position.***

Clearly the above sequence does not qualify (e.g., 100 follows 011 and these two strings differ in all three bit positions!).

Is there a way to generate all bit strings of length n such that every pair of adjacent strings differ in exactly one position?

The answer is yes!

Let’s start with 1 bit:

0

1

That was easy!

How about 2 bits? The regular numerical ordering doesn’t work (why?). How about this:

00

01

11

10

Let’s pick it apart:

0 0

0 1

1 1 Hey! the 2nd bit for the last two strings is

1 0 the same as for the first two, but just in reverse

order (top-to-bottom): 01 for the top-half and 10

for the bottom half!

Let’s see if we can make this a rule to get up to length 3:

Here is the grey code for length 2 again.

00

01

11

10

Now, let’s flip it (reverse the order top-to-bottom).

10

11

01

00

Now let’s stack them on top of each other and put a 0 in front of each string in the top half and a 1 in front of each string in the 2nd:

**0**00

**0**01

**0**11

**0**10

**1**10

**1**11

**1**01

**1**00

Cool! This follows the rules right?

**Your jobs:**

**First**: use the grey code for n=3 to construct a grey code of length 4 (with pencil and paper).

**Second**: complete a program which takes an integer from the command line and produces the corresponding grey code (all strings of that length in the correct order).

The C++ classes string and vector will probably come in handy! In particular, you can construct the codes as a vector of strings.

You have been given an incomplete program grey.cpp from which you can start (or you can start from scratch if you prefer).

Think recursively!!!

Reference: Need information about the vector class?

<http://www.cplusplus.com/reference/vector/vector/>