Odometer Problem

000000
000001
000002
…
000009
000010
…
000099
000100
…
999999

// Prints out all odometer settings where the first k items in array
// are fixed
Void odometer(int* array, int len, int k)

Odometer([3,2,6,5,9, x], 6, 5)

This does

326590
326591
326592
…
326599

Let's say we had 3 digits then
Odometer([2,1,0, x, x], 5, 3) would print

21000
21001
21002
21010
21011
21012
21020
21021
21022

Odometer([2,1,0, x, x], 5, 3)
    Odometer([2,1,0,0,x], 5, 4)
    Odometer([2,1,0,1,x], 5, 4)
    Odometer([2,1,0,2,x], 5, 4)

<u>What the pseudocode looks like…</u>

```
Odometer([2,1,0, x, x], length, k)
    For (d=0; d<NUMD; d++)
        Odometer([2,1,0,d,x], length, k+1)
```

Combinations

ABCDE

AB BC CD DE combos of size 2
AC BD CE
AD BE
AE

A B C D E combos of size 1

Empty Set

<u>Another View - each binary string corresponds to a combination</u>

ABCDE

	A	B	C	D	E
Empty	0	0	0	0	0
E	0	0	0	0	1
….					
BCE	0	1	1	0	1
…					
AC	1	0	1	0	0
…					

Permutation

ABC
ACB
BAC
BCA
CAB
CBA

Perm(int* array, int length, int* used, int k)

Perm([2, 4, 1, x, x, x], 6, [0, 1, 1, 0, 1, 0], 3)
2,4,1,0,3,5
2,4,1,0,5,3
2,4,1,3,0,5
2,4,1,3,5,0
2,4,1,5,0,3
2,4,1,5,3,0