Quick Review

N bits represent 2^N things:
How many bits do you need to represent 768 things?

Kind men give terminal pets extra zebra yolk:
2^{67} =

With 8 bits, what are the bit patterns for the following? For the last row, what is the decimal value of the given bit pattern?

	Unsigned	Sign & Magnitude	One's Complement	Two's Complement
`-1`				
`MAX`				
`MIN`				
`0x83`				

In general, with N bits the max/min for unsigned is _____, and for two's complement the max/min is _____.

What are the advantages and disadvantages of each integer representation?

Complete the following function `convert()` that takes an unsigned integer as an argument, and returns it's value when interpreted as a sign and magnitude number:

```
int convert(unsigned int signMag){




}
```

C details

```
int* p1, p2, p3, p4;
```
Did I just declare four pointers?

```
if ((5/4) * 100 == 125) printf("C can do math!\n");
```
Did it print?

Pointers

Writing the function swap and complete its call.

```
int foo = 5;
int baz = 42;
swap(       );
printf("foo is %d, baz is %d\n", foo, baz);
/* foo is 42, baz is 5 */
```

What is the output of the following program given this snapshot of memory?

Variable (if any)		a	b	c	p					x	y	
Address	...	171	172	173	174	175	176	177	...	655	656	...
Initial Value		15	19	-5	171	0	255	4		-1	8	

```
int main(int argc, char * argv[]){
      int a = 3, b = 144, c = 170;
      int *p;
      printf("%d, %d, %d\n", *p, p, &p);
      p = (int *) foo(a,&c);
      printf("%d, %d, %d\n", *p, p, &p);
      bar(&a, &b);
      printf("%d, %d, %d\n", a, b, c);
      return 0;
}
```

```
int foo (int x, int * y){
      *y = -12;
      return x + (int) y;
}

void bar (int * x, int * y){
      *x = *y;
      *y = (int) &y;
}
```

Bonus Question

What does this function do?

```
int mystery (unsigned int n)   {
   int count = 8 * sizeof(int) ;
   n ^= (unsigned int) - 1 ;
   while (n)  {
      count-- ;
      n &= (n - 1) ;
   }
   return count ;
}
```