Quick Review

N bits represent 2^N things:
How many bits do you need to represent 150 things? 8

Kind men give terminal pets extra zebra yolk:
2^{67} = 128 exbi

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`	N/A	1000 0001	1111 1110	1111 1111
`MAX`	1111 1111	0111 1111	0111 1111	0111 1111
`MIN`	0000 0000	1111 1111	1000 0000	1000 0000
`0x83`	131	-3	-124	-125

In general, with N bits the max/min for unsigned is _____2^N-1_____/_____0_____, and for two's
complement the max/min is _____2^(N-1) -1_____/_____-2^(N-1)_____.

What are the advantages and disadvantages of each integer representation?

-One's C: always increasing with "binary odometer", loops.
-Two's C: Only one zero.

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){
      if((1<<31)&signMag) { //if MSB is one
            return -(signMag & ~(1<<31)); // mask MSB to 0 and invert
      } else {
            return signMag;
      }
}
```

C details

```
int* p1, p2, p3, p4;
```

Did I just declare four pointers?
Sadly, this is interpretted as: int *p1, int p2, int p3, int p4. To declare the four pointers, we would need: int *p1, *p2, *p3, *p4.

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

Did it print?

The (5/4) is interpretted as integer division, and returns a value of one; thus, the statement does not print. We would require a conversion of one argument to floating point (ie (5/4.0) or (5/(double)4)), instead.

Pointers

Writing the function swap and complete its call.

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

```
void swap(int *a, int *b){
        int tmp = *a;
        *a = *b;
        *b = tmp;
}
```

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	
		3	144	170	176							
		144	656	-12								

```
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;
}
```

3, 171, 174
255, 176, 174
144, 656, -12