**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:
$2^{67} =$

**Fixed Width Numbers**

<table>
<thead>
<tr>
<th></th>
<th>Unsigned</th>
<th>Sign and Magnitude</th>
<th>One's Complement</th>
<th>Two's Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Def:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pros:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cons:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What is the value of the following bytes if interpreted as each of the above types?

0b0010 0010 0b1111 1111 0b1001 0111

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:

```c
int convert(unsigned int signMag) {
```

Klingon Word of the Week: ghun (gargle sound-oon) - to program (a computer)
Pointers

<table>
<thead>
<tr>
<th>Pointer Type</th>
<th>The Address Operator (&amp;)</th>
<th>The Dereference Operator (*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>int *x;</td>
<td>x = &amp;y;</td>
<td>*x = 5;</td>
</tr>
<tr>
<td>Tells the compiler to interpret the variable’s value as an address.</td>
<td>Returns the address of the variable provided. Used only for pointer assignment.</td>
<td>Essentially “follows” the pointer to allow for access to the referenced data.</td>
</tr>
</tbody>
</table>

Write a function that swaps the value of two integers:

```c
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;
}
```

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

<table>
<thead>
<tr>
<th>Variable (if any)</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>p</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>...</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
</tr>
<tr>
<td>Initial Value</td>
<td>15</td>
<td>19</td>
<td>-5</td>
<td>171</td>
<td>0</td>
<td>255</td>
</tr>
</tbody>
</table>

Klingon Word of the Week: ghun (gargle sound-oon) - to program (a computer)