

Bitwise Operations

1. For the following piece of code:

```
unsigned mystery5(unsigned x, unsigned p) {
    for (int result = 1; p != 0; p = p>>1) {
        if (p & 0x1) {
            result = result * x;
        }
        x = x*x;
    }
    return result;
}
```

a. Simulate it on $x = 2, p = 9$

"p" at the beginning of each for loop iteration	"result" at the end of the for loop iteration	"x" at the end of the for loop iteration
1001	$1*2 = 2$	$2*2 = 4$
0100	2	$4*4 = 16$
0010	2	$16*16 = 256$
0001	$2*256 = 512$	$256*256$
0000	$p == 0 \rightarrow$ return result = 512	

b. $x = 3, p = 4$

"p" at the beginning of each for loop iteration	"result" at the end of the for loop iteration	"x" at the end of the for loop iteration
0100	1	$3*3 = 9$
0010	1	$9*9 = 81$
0001	$1*81 = 81$	$81*81$
0000	$p == 0 \rightarrow$ return result = 81	

c. What does this function do? It returns the value of the arithmetic expression x^p (x to the pth power).

2. Fill in the blanks so that the following function in C **return the maximum integer in two's complement**, using only **integers between 1-32 (inclusive)** and **as few operators as possible**:

```
/** What should the parameter be? Fill in "void" if nothing is needed. */
```

```
int maxTwoC(void) {  
    return ~(1 << 31);  
}
```

3. Fill in the blanks so that the following function in C **returns 1 if x==y, otherwise 0**, using **as few operators as possible**:

```
int checkEqual(int x, int y) {  
    return !(x ^ y);  
}
```