Name: __Julia Nelson_____ Date: ___9/19/19_____

Pledge: _"I pledge my honor that I have abided by the Stevens Honor System."_____

For each function below, trace through it with reasonably small integer values. What does each function do?
HINT: You should assume integers are 8 bits for the purpose of this exercise.

```
int mystery1(int a, int b) {
    int c = a - b,
        d = (c >> 7) & 1,
        mystery = a - c * d;
    return mystery;
}
```

Trace: mystery1(3, 7) returns __7____(00000111)
Trace: mystery1(8, 7) returns __8____(00001000)

Summary: __ Returns the max of two integers.

```
void mystery2(int values[], int i, int j) {
    values[i] = values[i] ^ values[j];
    values[j] = values[i] ^ values[j];
    values[i] = values[i] ^ values[j];
}
```

Note: Improper C++ syntax found below.
Trace: mystery2([1, 2, 3, 4], 0, 3) values = [_4_, _2_, _3_, _1_]
Trace: mystery2([1, 2, 3, 4], 1, 2) values = [_1_, _3_, _2_, _4_]

Summary: __ Returns the values with values of indexes i and j swapped.

```
int mystery3(int x, int y) {
    int s, c;
    s = x ^ y;
    c = x & y;
    while (c != 0) {
        c = c << 1;
        x = s;
        y = c;
        s = x ^ y;
        c = x & y;
    }
    return s;
}
```

Trace: mystery3(5, 7) returns __12__(00001100)
Trace: mystery3(2, 8) returns __10__(00001010)

Summary: _Returns the sum of the two integers___

Lab 4: Bitwise & Bitshift Operators

① Trace:

Mystery1 (3,7) returns: 7

Mystery2 (8,7) returns: 8

(3,7) →

c = a-b = 3-7 = -4
d = (c >> 7) & 1

c: -4
-4: written 11111100
Shift7: 11111111 → -1
AND1: 00000001 → d = 1

mystery = a - c*d
= 3 - (-4)*1
= 7

(8,7) →

c = a-b = 8-7 = 1
d = (c>>7) & 1

c: 1
1: written 00000001
shift7: 00000000 → 0
AND1: 00000000 → d=0

mystery = a - c*d
= 8-(1)*0
= 8

Returns the Max of 2 ints

② Trace:

2.1) → mystery2 ([1,2,3,4], 0, 3) values =

2.2) → mystery2 ([1,2,3,4], 1, 2) values =

2.1) →

values[2] = 0101
values[3] = 0001
values[0] = 0100
v[0] = 5 → v[3] = 1 → v[2] = 4
[4,2,3,1]

values[0] = values[0] ^ values[3]
values[3] = values[0] ^ values[3]
values[0] = values[0] ^ values[3]

values = [4, 2, 3, 1]

2.2) →

v[1] = 0001
v[2] = 0010 (2)
v[3] = 0011 (3)
[1,3,2,4]

1 ≤ 2 ≤ 4

values[1] = values[1] ^ values[2] → 1234
values[2] = values[1] ^ values[2] → 1234
values[1] = values[1] ^ values[2] → 1324

values = [1, 3, 2, 4]

Returns values w/ values of index i and j swapped

③ Trace:

(5,7) → s = x^y → 0101 ^ 0111 = 0010 (2)
c = x & y → 0101 & 0111 = 0101 (5)
while c != 0
c << 1 → 0101 << 1 = 1010 (10) | 0100 (4)
x = s → 0010 (2) | 1000 (8)
y = c → 1010 (10) | 0100 (4)
s = x^y → 0010 ^ 1010 = 1000 (8) | 1000 ^ 0100 = 1100 ($2^3+2^2=12$)
c = x & y → 0010 & 1010 = 0010 (2) | 1000 & 0100 = 0000
return 1100 (12)

(2,8) → s = x^y → 0010 ^ 1000 = 1010 (10)
c = x & y → 0010 & 1000 = 0000 (0)
while c!=0: c << 1 → c = 0
x = s →
y = c →
s = x^y →
c = x & y →

return 1010 (10)