Bitwise Operations Test

a = ?
 	
int orig = 0xB0B0;
int insert = 0x000A;
int a = orig & (insert << 4);
	a = ?
	
	int orig = 0x2020;
	int insert = 0x0009;
	int a = orig ^ (insert << 8);

AND = ?
 	
int orig = 0x1010;
int insert = 0x0007;
int a = orig | (insert << 12)
int b = orig | (insert << 8)
int AND = a ^ b;
	AND = ?
	
	int orig = 0xC0C0;
	int insert = 0x0007;
	int a = orig | (insert << 4)
	int b = orig | (insert << 12)
	int AND = a ^ b;

result = ?
 	
long value1 = 0x80082002;
long value2 = 0x88002200;
int result = (value1 << 8) ^ (value2 >> 12);
	result = ?
	
	long value1 = 0x44008800;
	long value2 = 0x40048008;
	int result = (value1 << 12) ^ (value2 >> 8);

result = ?
 	
long value1 = 375;
long value2 = 883;
int result = (value1 << 12) ^ (value2 >> 4);
	result = ?
	
	long value1 = 352;
	long value2 = 169;
	int result = (value1 << 4) ^ (value2 >> 12);

right = ?
 	
int i = 0x1010;
int right = i & (1 << 8);
	a = ?
	
	long testValue = 0xCC00DD00;
	int a = 0;
	if (testValue & (1 << 8))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}

a = ?
 	
long testValue = 0x50059009;
int a = 0;
if (testValue ^ testValue | (1 << 12))
{
a = 1;
}
else
{
a = 2;
}
	a = ?, result = ?
	
	long testValue = 0xE00EB00B;
	int a = 0;
	if ((result = testValue ^ testValue | testValue
	& (1 << 4)))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}