Bitwise Operations Test

a = ?
 	
int orig = 0xB0B0;
int insert = 0x000F;
int a = orig ^ (insert << 8);
	a = ?
	
	int orig = 0x7070;
	int insert = 0x0007;
	int a = orig ^ (insert << 12);

OR = ?
 	
int orig = 0xF0F0;
int insert = 0x000D;
int a = orig | (insert << 8)
int b = orig | (insert << 4)
int OR = a | b;
	OR = ?
	
	int orig = 0x6060;
	int insert = 0x000D;
	int a = orig | (insert << 4)
	int b = orig | (insert << 12)
	int OR = a | b;

result = ?
 	
long value1 = 0xAA00FF00;
long value2 = 0xA00AF00F;
int result = (value1 << 12) ^ (value2 >> 4);
	result = ?
	
	long value1 = 0x70704040;
	long value2 = 0x70074004;
	int result = (value1 << 4) ^ (value2 >> 8);

result = ?
 	
long value1 = 33;
long value2 = 897;
int result = (value1 << 4) ^ (value2 >> 8);
	result = ?
	
	long value1 = 211;
	long value2 = 630;
	int result = (value1 << 4) ^ (value2 >> 8);

cupcake = ?
 	
int i = 0x7070;
int cupcake = i ^ (1 << 4);
	a = ?
	
	long testValue = 0x2200AA00;
	int a = 0;
	if (testValue & (1 << 4))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}

a = ?
 	
long testValue = 0x10105050;
int a = 0;
if (testValue & testValue ^ (1 << 4))
{
a = 1;
}
else
{
a = 2;
}
	a = ?, result = ?
	
	long testValue = 0x22002200;
	int a = 0;
	if ((result = testValue & testValue | testValue
	^ (1 << 4)))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}