Bitwise Operations Test

a = ?
 	
int orig = 0xA0A0;
int insert = 0x000D;
int a = orig & (insert << 4);
	a = ?
	
	int orig = 0xF0F0;
	int insert = 0x000F;
	int a = orig & (insert << 12);

OR = ?
 	
int orig = 0xF0F0;
int insert = 0x0007;
int a = orig | (insert << 4);
int b = orig | (insert << 8);
int OR = a ^ b;
	XOR = ?
	
	int orig = 0x9090;
	int insert = 0x0007;
	int a = orig | (insert << 4);
	int b = orig | (insert << 12);
	int XOR = a ^ b;

result = ?
 	
long value1 = 0xA00AD00D;
long value2 = 0xA0A0D0D0;
int result = (value1 << 8) ^ (value2 >> 12);
	result = ?
	
	long value1 = 0x2002A00A;
	long value2 = 0x2200AA00;
	int result = (value1 << 4) ^ (value2 >> 8);

result = ?
 	
long value1 = 937;
long value2 = 139;
int result = (value1 << 12) ^ (value2 >> 8);
	result = ?
	
	long value1 = 560;
	long value2 = 277;
	int result = (value1 << 8) & (value2 >> 12);

cupcake = ?
 	
int i = 0x3030;
int cupcake = i | (1 << 8);
	a = ?
	
	long testValue = 0x6006B00B;
	int a = 0;
	if (testValue & (1 << 4))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}

a = ?
 	
long testValue = 0xB00B2002;
int a = 0;
if (testValue | testValue ^ (1 << 4))
{
a = 1;
}
else
{
a = 2;
}
	a = ?, result = ?
	
	long testValue = 0x40402020;
	int a = 0;
	if ((result = testValue | testValue & testValue
	^ (1 << 4)))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}