Bitwise Operations Test

a = ?
 	
int orig = 0xB0B0;
int insert = 0x000C;
int a = orig | (insert << 12);
	a = ?
	
	int orig = 0xA0A0;
	int insert = 0x0009;
	int a = orig ^ (insert << 8);

XOR = ?
 	
int orig = 0x3030;
int insert = 0x0009;
int a = orig | (insert << 8)
int b = orig | (insert << 12)
int XOR = a ^ b;
	AND = ?
	
	int orig = 0x3030;
	int insert = 0x000E;
	int a = orig | (insert << 4)
	int b = orig | (insert << 8)
	int AND = a ^ b;

result = ?
 	
long value1 = 0xB0B0C0C0;
long value2 = 0xB00BC00C;
int result = (value1 << 4) ^ (value2 >> 12);
	result = ?
	
	long value1 = 0xA00AD00D;
	long value2 = 0xAA00DD00;
	int result = (value1 << 4) ^ (value2 >> 8);

result = ?
 	
long value1 = 46;
long value2 = 982;
int result = (value1 << 4) ^ (value2 >> 8);
	result = ?
	
	long value1 = 810;
	long value2 = 128;
	int result = (value1 << 8) ^ (value2 >> 4);

right = ?
 	
int i = 0x9090;
int right = i | (1 << 4);
	a = ?
	
	long testValue = 0x55009900;
	int a = 0;
	if (testValue & (1 << 12))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}

a = ?

```
long testValue = 0xF0F07070;
int a = 0;
if (testValue & testValue | (1 << 8))
{
a = 1;
}
else
{
a = 2;
}
```

a = ?, result = ?

```
long testValue = 0x77002200;
int a = 0;
if ((result = testValue ^ testValue & testValue
| (1 << 4)))
{
a = 1;
}
else
{
a = 2;
}
```