Bitwise Operations Test

a = ?
 	
int orig = 0xA406;
int insert = 0x0055;
int a = orig | (insert << 14);
	a = ?
	
	int orig = 0x7065;
	int insert = 0x00E2;
	int a = orig | (insert << 15);

OR = ?
 	
int orig = 0xD60D;
int insert = 0x0038;
int a = orig | (insert << 7)
int b = orig | (insert << 2)
int OR = a | b;
	AND = ?
	
	int orig = 0x8AAE;
	int insert = 0x007F;
	int a = orig | (insert << 7)
	int b = orig | (insert << 11)
	int AND = a | b;

result = ?
 	
long value1 = 0xF8EBF897;
long value2 = 0x92A8B944;
int result = (value1 << 2) ^ (value2 >> 6);
	result = ?
	
	long value1 = 0xED9C749B;
	long value2 = 0xE524AE8D;
	int result = (value1 << 11) ^ (value2 >> 14);

result = ?
 	
long value1 = 12418;
long value2 = 6549;
int result = (value1 << 1) ^ (value2 >> 5);
	result = ?
	
	long value1 = 8832;
	long value2 = 12641;
	int result = (value1 << 13) ^ (value2 >> 15);

right = ?
 	
int i = 0x705C;
int right = i | (1 << 11);
	a = ?
	
	long testValue = 0x90F00010;
	int a = 0;
	if (testValue & (1 << 1))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}

a = ?
 	
long testValue = 0x37006200;
int a = 0;
if (testValue ^ testValue & (1 << 13))
{
a = 1;
}
else
{
a = 2;
}
	a = ?, result = ?
	
	long testValue = 0xBA000A00;
	int a = 0;
	if ((result = testValue ^ testValue & testValue
	| (1 << 14)))
	{
	a = 1;
	}
	else
	{
	a = 2;
	}