Bitwise Operations Test

<table>
<tr>
<td>a = ?

int orig = 0x793F;
int insert = 0x00AB;
int a = orig | (insert << 7);</td>
<td>a = ?

int orig = 0xCDC3;
int insert = 0x00E8;
int a = orig ^ (insert << 9);</td>
</tr>
<tr>
<td>XOR = ?

int orig = 0xDB9F;
int insert = 0x008C;
int a = orig | (insert << 7)
int b = orig | (insert << 9)
int XOR = a ^ b;</td>
<td>XOR = ?

int orig = 0xF0A1;
int insert = 0x0096;
int a = orig | (insert << 3)
int b = orig | (insert << 1)
int XOR = a | b;</td>
</tr>
<tr>
<td>result = ?

long value1 = 0xB561AD43;
long value2 = 0x95A5D48A;
int result = (value1 << 7) ^ (value2 >> 3);</td>
<td>result = ?

long value1 = 0x41751267;
long value2 = 0xD5EF9501;
int result = (value1 << 1) ^ (value2 >> 3);</td>
</tr>
<tr>
<td>result = ?

long value1 = 4748;
long value2 = 12179;
int result = (value1 << 5) ^ (value2 >> 6);</td>
<td>result = ?

long value1 = 15;
long value2 = 1169;
int result = (value1 << 15) ^ (value2 >> 13);</td>
</tr>
<tr>
<td>left = ?

int i = 0x0844;
int left = i | (1 << 2);</td>
<td>a = ?

long testValue = 0x06000D00;
int a = 0;
if (testValue & (1 << 1))
{
a = 1;
}
else
{
a = 2;
}</td>
</tr>
</table>

<table>
<tr><td></td><td></td></tr>
<tr><td>a = ?

long testValue = 0xC1006F00;
int a = 0;
if (testValue & testValue | (1 << 11))
{
a = 1;
}
else
{
a = 2;
}</td><td>a = ?, result = ?

long testValue = 0x700CB006;
int a = 0;
if ((result = testValue ^ testValue & testValue
| (1 << 11)))
{
a = 1;
}
else
{
a = 2;
}</td></tr>
</table>