What is the value (all 8 hex digits) of r3 after each set of instructions:

```assembly
ldr r1, =0x98765432
ldr r2, =0x2468ACDC
and r3, r1, r2
```

Value in r3 at this point is \( 0x \)___________________________

```assembly
ldr r1, =0x98765432
ldr r2, =0x2468ACDC
eor r3, r1, r2
```

Value in r3 at this point is \( 0x \)___________________________

```assembly
ldr r1, =0x98765432
asr r3, r1, 8
```

Value in r3 at this point is \( 0x \)___________________________

Fill in the blanks to correctly implement the following C code fragment in ARM assembly. Assume param1 and param2 are formal parameters available in the first two parameter registers. Assume x is mapped to register r3.

```c
if ( param1 > param2 ) {
    x = 44;
} else {
    x = 22;
}
```

Which of the logic gate symbols represents … … OR … AND … XOR … NAND

```
    mov    r3, 44  @ x = 44;
    mov    r3, 22  @ x = 22;
```
Write the equivalent **unoptimized** ARM assembly language instructions to perform the following C code fragment. **Use the loop construct specified in class/Notes.**

```c
a = x;
while ( a != 25 ) {
    a = --x + 9;
}
```

<table>
<thead>
<tr>
<th>ARM assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>/* x is mapped to r4 */</td>
</tr>
<tr>
<td>/* a is mapped to r5 */</td>
</tr>
</tbody>
</table>

Write the equivalent **unoptimized** ARM assembly language instructions to perform the following C code fragment. Use magic numbers vs. defining an assembler constant.

```c
x = foo( 42, 420420 );
```

<table>
<thead>
<tr>
<th>ARM assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>/* x is mapped to r4 */</td>
</tr>
</tbody>
</table>

Write the ARM assembly instructions for the following global variable definition:

```c
short bar = 42;
```