Bit operations

Email if you have any questions!

☐ I pledge my honor that I have abided by the Stevens Honor System Turn audio ON

Time Remaining:

-1:57 Submit Quiz

1. bitwise operations (1 points) Click to report a problem

```
Write the answers in hexidecimal
mov r0, #27       @ r0=[0000001B]
mov r1, #105      @ r1=[00000069]
mov r2, #92       @ r2=[0000005C]
and r3, r1, r2    @ r3=[00000048]
orr r4, r1, r0    @ r4=[0000007B]
eor r5, r4, r1    @ r5=[00000012]
mvn r0, r5        @ r5=[FFFFFFED]
```

2. (1 points) Click to report a problem

```
mov r1, #13
mov r0, r1
lsl r0, #4    @r0 = [000000D0]
mov r1, #8
orr r0, r1    @r0 = [000000D8]
lsl r0, #4    @r0 = [00000D80]
orr r0, r1    @r0 = [00000D88]
mov r1, #14
eor r0, r1    @r0 = [00000D86]
```

3. Odd or even (1 points) Click to report a problem

```
Complete the following code so it jumps to odd if the rightmost bit of r0 is set

oddoreven:
            [tst] r0, #1
            [bne] odd
```

4. Set the middle bits (1 points) Click to report a problem

```
The first instruction loads a hex number with a zero in it.
Using or and shifting, replace the 0 with 9
ldr   r0, =0xfab903dc
mov   r1, #9
[LSL] r1, #12
[ORR] r0, [R1]
```

5. clear the middle bits (1 points) Click to report a problem

```
The first instruction loads a hex number with a zero in it.
Using and and shifting, replace the f with a
ldr   r0, =0x34bf23dc
ldr   r1, =0xFFFAFFFF
AND   r0, r1
```

6. Replace the nibble (1 points) Click to report a problem

```
The first instruction loads a hex number with a 5 in it.
Replace it with 9. To do this, clear the bits
and write new ones with OR

ldr   r0, =0x34bf235c
mov   r1, #f
LSL   r1, #4
BIC   r0, r1
mov   r1, #0x90
ORR   r0, r1
```

7. Now do it with 6 bits (1 points) Click to report a problem

```
This problem is similar, but now the number of bits is different so you will not
be changing only a single hex digit. The first instruction loads a number.
Given that bit 0 is the rightmost bit, replace bits 12-17 with the value 49.

ldr   r0, =0x12345678
mov   r1, #0x3F     @ figure out how to write 6 bits in hex
LSL   r1, #12  @ shift it into position
BIC   r0, r1        @ clear out the desired bits
mov   r1, #49           @ load in the new, desired number
ORR   r0, r1        @ write in the new bits
```

8. Unix file permissions (1 points) Click to report a problem

```
In Unix, files have permissions read (r), write (w) and execute (x)
A file has an owner and a group. The basic permissions take 9 bits.
The first 3 describe what the owner of the file can do.
The second 3 describe what anyone in the same group can do.
The last 3 describe what anyone on the computer may do.
For example, given permissions:
rw-r--r-- dkruger tomcat  myfile.txt
The file myfile.txt may be read and written by owner dkruger, read by anyone in th
and read by anyone else on the computer. The corresponding bits are:
110100100

Given the above permissions are set, write ARM assembler instructions to remove th
write for the owner, remove the right to read for everyone (leaving the group), an
to execute for the owner. The resulting bits should be:
101100000

ldr   r0, =0x01A4       @ load initial permissions
mov   r1, #0x01FF     @ load a single mask to clear the desired bits
bic   r0, r1            @ clear out the desired bits
```

```
mov   r1, #0x160    @ set up second mask to write in 1 bits
ORR  r0, r1       @ write in the new bits
```

9. Test bits = val (1 points) Click to report a problem

Complete the following code so it jumps to yes if the bits in r0 from 4 to 7 = 100
In other words, if r0 = 0xfa120891 it should jump to "yes" because 9 = 1001

```
oddoreven:
    lsr    r0, #4
    and    r0, #15
    cmp    r0, #9
    beq  yes
```

Time Remaining: -1:57 Submit Quiz