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, #0x3a
mov r1, #97       @ r1= 00000061
mov r2, #161      @ r2= 000000A1
and r3, r1, r2    @ r3= 00000021
orr r4, r1, r0    @ r4= 0000007B
eor r5, r4, r1    @ r5= 0000001A
mvn r0, r5        @ r5= FFFFFFE5
eor r0, r0        @ r0= 00000000
```

2. (1 points)

Click to report a problem

```
mov r1, #15
mov r0, r1
lsl r0, #4    @r0 = 000000F0
mov r1, #9
orr r0, r1    @r0 = 000000F9
lsl r0, #5    @r0 = 00001F20
orr r0, r1    @r0 = 00001F29
mov r1, #14
eor r0, r1    @r0 = 00001F27
```

3. Odd or even (1 points)

Click to report a problem

```
Complete the following code so it executes the code at
odd if the rightmost bit of r0 is set, and even if it is not

oddoreven:
        mov     r0, #23
        tst     r0, #1
        bne     even
odd:
        mov     r1, #17
        b       1f
even:
        mov     r1, #4
1:

At the end of the code (at label 1:), what is the value of r1?
```

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 00 with AA
```

```
ldr   r0, =0xfab900dc
mov   r1, #0xAA
lsl   r1, ##8
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 F8 with 00
ldr   r0, =0x34BF83DC
ldr   r1, =0xFFF00FFF
AND   r0, r1
```

6. clear the middle bits using BIC (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 F8 with 00
ldr   r0, =0x34BF83DC
ldr   r1, =0xFFFF00FF
and   r0, r1
```

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

```
The first instruction loads a hex number with a 2 in it.
Replace it with A. To do this, shift the mask in r1 to the left,
clear the bits, and write new ones using OR

ldr   r0, =0x34bf235c
mov   r1, #f
lsl   r1, #12
bic   r0, r1
mov   r1, #0xA
lsl   r1, #12
orr   r0, r1
```

8. 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 14-19 with the value 27.

ldr   r0, =0x12345678
mov   r1, #0xFC        @ 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, #27           @ load in the new, desired number
lsl   r1, #12       @ shift it into position
orr   r0, r1        @ write in the new bits
```

9. 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------- 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:
110000000

Given the above permissions are set, write ARM assembler instructions to remove th
write for everyone. Then add the right to read for everyone. The resulting bits sh
10010100

ldr   r0, =0x0180       @ load initial permissions
mov   r1, #0x0100      @ load a single mask to clear the desired bits
bic   r0, r1            @ clear out the desired bits
ldr   r1, =0xA8        @ set up second mask to write in 1 bits
orr  r0, r1      @ write in the new bits
```

Time Remaining: -1:57

Submit Quiz