

Name: Tran Duc Anh Dang

8.1

The screenshot shows the ARM Lite Simulator V1.2 interface. The Program window contains the following assembly code:

```
0x00000000: mov r0, #15
2: str r0, .WriteSignedNum
3: mov r1, #dp1
4: str r1, .WriteString
5: mov r2, #dp2
6: str r2, .WriteString
7: mov r3, #num
8: ldr r3, .inputNum
9: mov r4, #dp3
10: str r4, .WriteString
11: sub r0, r0, r3
12: str r0, .WriteSignedNum
13: halt
14 dp1: .asciz "remaining\n"
15 dp2: .asciz "How many do you want to remove (1-3)?\n\n"
16 num: .word 0
17 dp3: .asciz "The remaining match sticks are "
```

The Processor window shows the PC at 1048576, the instruction at 00000000, and the status bits at 0000. The Memory window shows the address 00000000. The Input/Output window shows the prompt "How many do you want to remove (1-3)?".

8.2

The screenshot shows the ARM Lite Simulator V1.2 interface after the user has entered a value. The Program window contains the same assembly code as in the previous screenshot. The Processor window shows the PC at 1048576, the instruction at 00000000, and the status bits at 0000. The Memory window shows the address 00000000. The Input/Output window shows the prompt "How many do you want to remove (1-3)?" and the user's input "3".

If you enter a number that takes the number of matchsticks remaining beyond 0 (into negative values), the result will go into negative value in r0. It represented using 2's compliment principle.

8.2.2(a) - What is the condition that needs to be satisfied in order for this loop to occur? Write this as a comparison using an inequality (ie., less than, greater than, less than or equal, greater than or equal)

- The value must be entered between 1-3.

Question 8.2.2(b) - What two ARM assembly instructions could be used to create a branch that only occurs under this condition?

- CMP, BLT and BGT

Question 8.2.2(c) - Based on the instructions you outlined in 8.2.2(b), what status bit would be set to 1 if the loop was to repeat ?

- N and Z bit should be set to 1 if the instruction is executed. Otherwise, when the entered value is less than 3, only N bit is set, as we compared the entered value with 3, so the result in this case will obviously negative.

Question 8.2.2(d) - What are all the modifications needed to the current program to implement this feature ? Make the required modifications to your program to perform the task.

The screenshot shows the ARMLite Simulator interface. The **Program** panel on the left contains the following code:

```

1: mov r0, #15
2: Loop:
3:   str r0, .WriteSignedNum
4:   mov r1, #dpl
5:   str r1, .WriteString
6:   mov r2, #dpl
7:   Loop2:
8:     str r2, .WriteString
9:     mov r3, #num
10:    ldr r3, .inputNum
11:    cmp r3, #1
12:    btl Loop2
13:    b condition1
14: condition1:
15:    cmp r3, #3
16:    bgt Loop2
17:    b condition2
18: condition2:
19:    sub r0, r0, r3
20:    cmp r0, #0
21:    bgt Loop
22:    mov r5, #dpl3
23:    str r5, .WriteString
24:    halt
25: dpl1: .ascii "remaining\n"
26: dpl2: .ascii "How many do you want to remove (1-3)?\n\n"
27: dpl3: .ascii "There are none left!"
28: num: .word 0
  
```

The **Processor** panel shows the PC at 1048576, SP at 1048576, and various registers. The **Memory** panel shows a memory dump starting at 0x0000. The **Input/Output** panel shows a message box asking "How many do you want to remove (1-3)?".

8.3.1(a) - What bit-wise operation can we perform on the register holding the 32 bit pattern to set all bits in the register to zero except the least significant 2 bits ? Write this as a single line of code.

- That bitwise operation is AND, when we will use that operation with the random number and 0b00000000000000000000000000000011

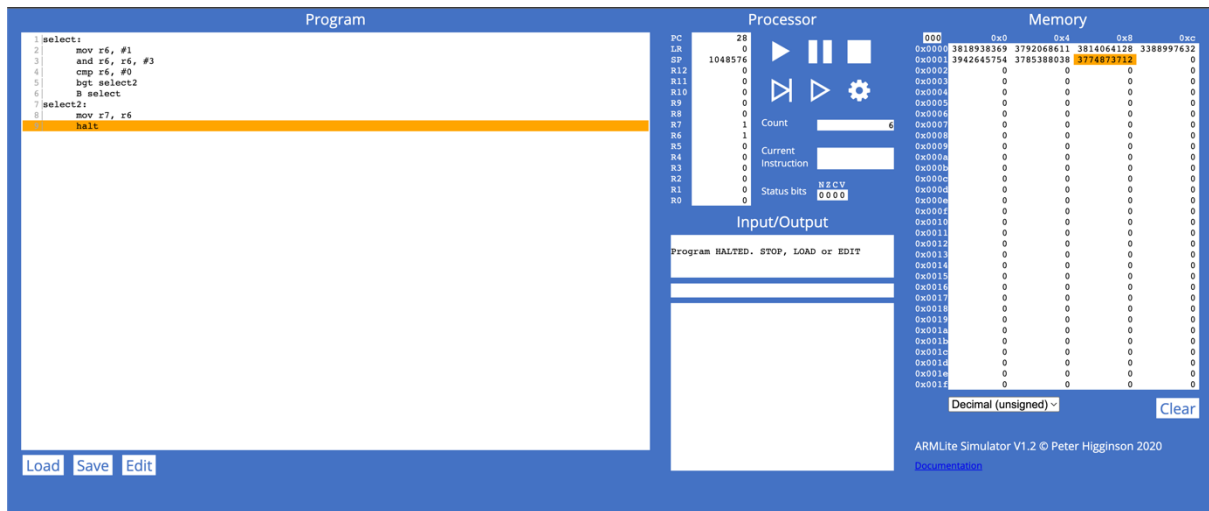
Question 8.3.1(b) Using a label named "select:" Write the code needed to repeatedly sample a random number (from .Random) until the value is in the range 1-3. For now, just write this as a separate program and test it.

The screenshot shows the ARMLite Simulator interface. The **Program** panel on the left contains the following code:

```

21: cmp r4, #0
22: bgt select
23: R0B R0, #0, #6
24: R0B R0, #0
25: R0B prompt1
26: R0B Loop
27: prompt1:
28: mov r12, #dpl
29: str r12, .WriteString
30: halt
31: prompt2:
32: mov r12, #dpl
33: str r12, .WriteString
34: halt
35: Loop1:
36: str r2, .WriteString
37: mov r3, #num
38: ldr r3, .inputNum
39: cmp r3, #1
40: btl Loop1
41: b condition1
42: condition1:
43: cmp r3, #3
44: bgt Loop1
45: b condition2
46: condition2:
47: sub r0, r0, r3
48: cmp r0, #0
49: bgt prompt2
50: bgt select
51: mov r5, #dpl3
52: str r5, .WriteString
53: halt
54: dpl1: .ascii "remaining\n"
55: dpl2: .ascii "How many do you want to remove (1-3)?\n\n"
56: dpl3: .ascii "There are none left!\n"
57: dpl4: .ascii "It's a computer's turn!!\n"
58: dpl5: .ascii "You Win!!\n"
59: dpl6: .ascii "You Lose!!\n"
60: num: .word 0
  
```

The **Processor** panel shows the PC at 112, SP at 1048576, and various registers. The **Memory** panel shows a memory dump starting at 0x0000. The **Input/Output** panel shows a message box asking "How many do you want to remove (1-3)?".



Code:

```

1|  mov r0, #15
2| Loop:
3|  str r0, .WriteSignedNum
4|  mov r1, #dp1
5|  str r1, .WriteString
6|  mov r2, #dp2
7|  B Loop2
8| select:
9|  str r0, .WriteSignedNum
10|  mov r1, #dp1
11|  str r1, .WriteString
12|  mov r7, #dp4
13|  str r7, .WriteString
14| select_again:
15|  LDR R6, .Random
16|  AND R6, R6, #3
17|  CMP R6, #0
18|  BGT select2
19|  B select_again
20| select2:
21|  CMP R6, R0
22|  BGT select
23|  SUB R0, R0, R6
24|  CMP R0, #0
25|  BEQ prompt1
26|  BGT Loop
27| prompt1:
28|  mov r12, #dp5
29|  str r12, .WriteString
30|  halt
31| prompt2:
32|  mov r12, #dp6
33|  str r12, .WriteString

```

```

34|    halt
35|Loop2:
36|    str r2, .WriteString
37|    mov r3, #num
38|    ldr r3, .InputNum
39|    cmp r3 , #1
40|    blt Loop2
41|    b condition1
42|condition1:
43|    cmp r3, #3
44|    bgt Loop2
45|    b condition2
46|condition2:
47|    sub r0 , r0 , r3
48|    cmp r0 , #0
49|    beq prompt2
50|    bgt select
51|    mov r5, #dp3
52|    str r5, .WriteString
53|    halt
54|dp1: .asciz "remaining\n"
55|dp2: .asciz "How many do you want to remove (1-3)?\n\n"
56|dp3: .asciz "There are none left!\n"
57|dp4: .asciz "____!!!It's computer's turn!!!____\n"
58|dp5: .asciz "\nYou Win!!! \n\n "
59|dp6: .asciz "\nYou Loose!!!\n\n"
60|num: .word 0

```