

Lab08

Nguyen Nam Tung 103181157

8.1.

Program

```
1|    mov r0, #15
2|    str r0, .WriteSignedNum
3|    mov r1, #ms1
4|    str r1, .WriteString
5|    mov r2, #ms2
6|    str r2, .WriteString
7|    mov r3, #num
8|    ldr r3, .InputNum
9|    mov r4, #ms3
10|   str r4, .WriteString
11|   sub r0, r0, r3
12|   str r0, .WriteSignedNum
13|   halt
14| ms1: .asciz "remaining\n"
15| ms2: .asciz "How many do you want to remove (1-3)\n"
16| num: .word 0
17| ms3: .asciz "The remaining match stick are "
```

Load Save Edit

Processor

PC	0x00000034
LR	0x000ffffc
SP	0x00100000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0xfe696e69
R5	0x00000000
R4	0x0000006c
R3	0x00000001
R2	0x0000003f
R1	0x00000034
R0	0x0000000e

Count 129761

Current Instruction

Status bits NZCV 0000

Input/Output

remaining
How many do you want to remove (1-3)
The remaining match stick are 14
Program HALTED. STOP, LOAD or EDIT

1

8.2.

The screenshot shows an ARM assembly simulator interface. On the left, a list of assembly instructions is displayed with line numbers 1 through 16. The instructions include moving values to registers, writing strings, and a loop structure. On the right, a panel shows the current state of the processor: registers R0 through R15, the Program Counter (PC), and status bits (N, Z, C, V). Below the register list, there are controls for execution (play, pause, step, reset) and a 'Count' field set to 116. At the bottom right, an 'Input/Output' section shows a text area with the prompt 'How many do you want to remove (1-3)' and the output '3 remaining'. Below this is an 'Input expected' field.

```

1  mov r0, #15
2  str r0, .WriteSignedNum
3  Loop:
4      mov r1, #ms1
5      str r1, .WriteString
6      mov r2, #ms2
7      str r2, .WriteString
8      mov r3, #num
9      ldr r3, .InputNum
10     sub r0, r0, r3
11     str r0, .WriteSignedNum
12     B Loop
13     halt
14 ms1: .asciz "remaining\n"
15 ms2: .asciz "How many do you want to remove (1-3)\n"
16 num: .word 0

```

Registers and PC values:

- PC: 0x00000020
- LR: 0x00000000
- SP: 0x00100000
- R12: 0x00000000
- R11: 0x00000000
- R10: 0x00000000
- R9: 0x00000000
- R8: 0x00000000
- R7: 0x00000000
- R6: 0x00000000
- R5: 0x00000000
- R4: 0x00000000
- R3: 0x00000064
- R2: 0x0000003b
- R1: 0x00000030
- R0: 0x00000003

Status bits: N Z C V 0 0 0 0

Count: 116

Current Instruction: [Empty]

Input/Output:

How many do you want to remove (1-3)
3 remaining
How many do you want to remove (1-3)

Input expected: [Empty]

The values entered must be between 1 and 3

⇒ We can use CMP and BLT

8.2.2

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.

Full Code

```
mov r0, #15
```

```
Loop:
```

```
    str r0, .WriteSignedNum
```

```
    mov r1, #ms1
```

```
    str r1, .WriteString
```

```
    mov r2, #ms2
```

```
    B Loop2
```

```
select:
```

```
    str r0, .WriteSignedNum
```

```
    mov r1, #ms1
```

```
str r1, .WriteString
mov r7, #ms4
str r7, .WriteString
```

select_again:

```
LDR R6, .Random
AND R6, R6, #3
CMP R6, #0
BGT select2
B select_again
```

select2:

```
CMP R6, R0
BGT select
SUB R0, R0, R6
CMP R0, #0
BEQ prompt1
BGT Loop
```

prompt1:

```
mov r12, #ms5
str r12, .WriteString
halt
```

prompt2:

```
mov r12, #ms6
str r12, .WriteString
halt
```

Loop2:

```
str r2, .WriteString
mov r3, #num
ldr r3, .InputNum
cmp r3, #1
```

blt Loop2

b condition1

condition1:

cmp r3, #3

bgt Loop2

b condition2

condition2:

sub r0, r0, r3

cmp r0, #0

beq prompt2

bgt select

mov r5, #ms3

str r5, .WriteString

halt

ms1: .asciz "remaining\n"

ms2: .asciz "How many do you want to remove (1-3)?\n\n"

ms3: .asciz "There are no sticks left!\n"

ms4: .asciz "____Computer Turn____\n"

ms5: .asciz "\nWin \n\n "

ms6: .asciz "\nLose\n\n"

num: .word 0