

Exercise 8.1.1 – 8.1.3

Program

```

1  MOV R0, #15
2  STR R0, .WriteUnsignedNum
3  MOV R1, #msg1
4  STR R1, .WriteString
5  MOV R1, #msg2
6  STR R1, .WriteString
7  LDR R2, .InputNum
8  SUB R0,R0,R2
9  STR R0, .WriteUnsignedNum
10 MOV R1, #msg1
11 str r1, .WriteString
12 HALT
13 msg1: .ASCIIZ "remaining\n"
14 msg2: .ASCIIZ "How many do you want

```

Processor

PC	44
LR	0
SP	1048576
R12	0
R11	0
R10	0
R9	0
R8	0
R7	0
R6	0
R5	0
R4	0
R3	0
R2	7
R1	48
R0	8

Count: 11
Current: e50f1110
Instruction: STR Rd,addr
Status bits: NZCV 0000

Input/Output

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

Done instruction STR Rd,addr at line 11

7

Exercise 8.2.1

Program

```

1  MOV R0, #15
2  STR R0, .WriteUnsignedNum
3  MOV R1, #msg1
4  STR R1, .WriteString
5  Loop:
6  MOV R1, #msg2
7  STR R1, .WriteString
8  LDR R2, .InputNum
9  SUB R0,R0,R2
10 STR R0, .WriteUnsignedNum
11 MOV R1, #msg1
12 STR R1, .WriteString
13 B Loop
14 HALT
15 msg1: .ASCIIZ "remaining\n"
16 msg2: .ASCIIZ "How many do you want

```

Processor

PC	28
LR	0
SP	1048576
R12	0
R11	0
R10	0
R9	0
R8	0
R7	0
R6	0
R5	0
R4	0
R3	0
R2	1
R1	63
R0	4294967295

Count: 55
Current:
Instruction:
Status bits: NZCV 0000

Input/Output

How many do you want to remove (1-3)?
4294967295 remaining
How many do you want to remove (1-3)?

Input expected

If we enter a number beyond 0, let's say we want to remove 1 when there are 0 matchsticks left. It stills works normally as the actual value in register shows -1, meanwhile the register shows a big number in unsigned decimal.

Exercise 8.2.2

- $0 < R2 < 4$
- Two assembly instructions that can be used are: BGT and BLT
BGT: Z clear, N and V are the same.
BLT: N and V different
- If the first condition is not met ($R2 > 0$), and R2 is negative, then N = 1. If the first condition is not met ($R2 > 0$), and R2 = 0 then Z = 1
If the second condition is not met ($R2 < 4$), and R2 > 4 then C = 1. If the second condition is not met ($R2 < 4$), and R2 = 4 then both Z = 1 and C = 1.

```

(d) 1|  MOV R0, #15
    2|  STR R0, .WriteUnsignedNum
    3|  MOV R1, #msg1
    4|  STR R1, .WriteString
    5| Loop:
    6|  MOV R1, #msg2
    7|  STR R1, .WriteString
    8|  LDR R2, .InputNum
    9| start:
   10|  CMP R2, #0
   11|  BGT else1      //If R2 > 0 jump to else1
   12|  B invalid1
   13| else1:
   14|  CMP R2, #4
   15|  BLT cont      //If R2 < 4 jump to cont
   16|  B invalid1
   17| invalid1:
   18|  MOV R1, #msg3
   19|  STR R1, .WriteString
   20|  LDR R2, .InputNum
   21|  B start
   22| cont:
   23|  SUB R0,R0,R2
   24|  STR R0, .WriteUnsignedNum
   25|  MOV R1, #msg1
   26|  STR R1, .WriteString
   27|  B Loop
   28|  HALT
   29| msg1: .ASCIZ "remaining\n"
   30| msg2: .ASCIZ "How many do you want to remove (1-3)?\n"
   31| msg3: .ASCIZ "Please input a valid number\n"

```

The screenshot displays a program execution simulator with three main panels:

- Program:** Shows the assembly code from the previous block, with line numbers 1 through 31. The code includes instructions for moving values, storing to memory, comparing, branching, and looping.
- Processor:** Displays the current state of the processor.
 - PC (Program Counter): 64
 - LR (Link Register): 0
 - SP (Stack Pointer): 1048576
 - Registers R12 through R0: All are 0 except R2 (4), R1 (142), and R0 (10).
 - Count: 48
 - Current Instruction: (empty)
 - Status bits: NZCV 0110
- Input/Output:** Shows the output of the program.
 - 10 remaining
 - How many do you want to remove (1-3)?
 - Please input a valid number
 - Input expected (in a text box)

Exercise 8.3.1




- (a) LSL R4, R4, #30
LSR R4,R4, #30
- (b) Select:
LDR R4, .Random
LSL R4,R4, #30
LSR R4,R4, #30
CMP R4, #0
BGT conti
B select
Conti:
STR R4, .WriteUnsignedNum
HALT




Program

```
1 select:
2     LDR R4, .Random
3     LSL R4,R4, #30
4     LSR R4,R4, #30
5     CMP R4, #0
6     BGT conti
7     B select
8 conti:
9     STR R4, .WriteUnsignedNum
10    HALT
```

Processor

PC	0x00000020
LR	0x00000000
SP	0x00100000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0x00000000
R5	0x00000000
R4	0x00000001
R3	0x00000000
R2	0x00000000
R1	0x00000000
R0	0x00000000





Count

Current Instruction

Status bits

N	Z	C	V
0	0	0	0

Input/Output

1
Program HALTED. STOP, LOAD or EDIT

Exercise 8.3.2

- MOV R0, #3
- Select:
- LDR R4, .Random
LSL R4, R4, #30
LSR R4,R4, #30
CMP R4, #0
BGT conti
B select
- Conti:

CMP R4, R0

BGT select

B continue

Continue:

STR R4, .WriteUnsignedNum

HALT

Program

```
1  MOV R0, #3
2  select:
3      LDR R4, .Random
4      LSL R4, R4, #30
5      LSR R4, R4, #30
6      CMP R4, #0
7      BGT conti
8      B select
9  conti:
10     CMP R4, R0
11     BGT select
12     B continue
13 continue:
14     STR R4, .WriteUnsignedNum
15     HALT
```

Processor

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

Count

Current Instruction

Status bits **N Z C V**
0 1 1 0

Input/Output

3

Program HALTED. STOP, LOAD or EDIT

Exercise 8.4.1 – 8.4.2

```
1 //Initialise matchstick count
2 MOV R0, #15
3 Loop:
4     STR R0, .WriteUnsignedNum
5 //Display initial message
6     MOV R1, #msg1
7     STR R1, .WriteString
8 //Computers Turn
9 select: LDR R2, .Random
10     AND R2, R2, #3
11     CMP R2, #0
12     BEQ select
13     CMP R2, R0
14     BGT select
15     BEQ select
16 cont: STR R2, .WriteUnsignedNum
17     MOV R1, #msg4
18     STR R1, .WriteString
19     SUB R0, R0, R2
20 //Print remaining matchstick
21     STR R0, .WriteUnsignedNum
22     MOV R1, #msg1
23     STR R1, .WriteString
24 //Check if computer wins
25     cmp r0, #1
26     BEQ computerWins
27 //Players turn
28     MOV R1, #msg2
29     STR R1, .WriteString
30 input: LDR R2, .InputNum
31     CMP R2, #3
32     BGT input
33     CMP R2, #1
34     BLT input
35     CMP R2, R0
36     BGT input
37     SUB R0, R0, R2
38     CMP R0, #1
39     BEQ playerWins
```

PC	0x00000090
LR	0x00000000
SP	0x00100000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0x00000000
R5	0x00000000
R4	0x00000000
R3	0x00000000
R2	0x00000002
R1	0x0000000ce
R0	0x000000001

Count

Current Instruction

Status bits **N Z C V**
0 1 1 0

Input/Output

3 remaining

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

You Win!

Program HALTED. STOP, LOAD or EDIT

2

Load Save Edit

Program

```

13|    AND R2,R0
14|    BGT select
15|    BEQ select
16|cont: STR R2, .WriteUnsignedNum
17|    MOV R1, #msg4
18|    STR R1, .WriteString
19|    SUB R0, R0, R2
20|    //Print remaining matchstick
21|    STR R0, .WriteUnsignedNum
22|    MOV R1, #msg1
23|    STR R1, .WriteString
24|    //Check if computer wins
25|    cmp r0, #1
26|    BEQ computerWins
27|    //Players turn
28|    MOV R1, #msg2
29|    STR R1, .WriteString
30|input: LDR R2, .InputNum
31|    CMP R2, #3
32|    BGT input
33|    CMP R2, #1
34|    BLT input
35|    CMP R2,R0
36|    BGT input
37|    SUB R0,R0,R2
38|    CMP R0,#1
39|    BEQ playerWins
40|    b Loop
41|playerWins: MOV R1, #msg3
42|    STR R1, .WriteString
43|    HALT
44|computerWins: MOV R1, #msg5
45|    STR R1, .WriteString
46|    HALT
47|msg1: .ASCIIZ "remaining\n"
48|msg2: .ASCIIZ "How many do you want to remove (1-3)?\n"
49|msg3: .ASCIIZ "You Win!"
50|msg4: .ASCIIZ "taken by computer\n"
51|msg5: .ASCIIZ "Computer Wins!\n"

```

Load

Save

Edit

Processor

PC	0x00000090
LR	0x00000000
SP	0x00100000
R12	0x00000000
R11	0x00000000
R10	0x00000000
R9	0x00000000
R8	0x00000000
R7	0x00000000
R6	0x00000000
R5	0x00000000
R4	0x00000000
R3	0x00000000
R2	0x00000002
R1	0x000000ce
R0	0x00000001

▶

⏸

□

⏮

⏭

⚙

Count

116

Current Instruction

Status bits

NZCV 0110

Input/Output

3 remaining
How many do you want to remove (1-3)?
You Win!
Program HALTED. STOP, LOAD or EDIT

2

```

1|//Initialise matchstick count
2|    MOV R0, #15
3|Loop:
4|    STR R0, .WriteUnsignedNum
5|//Display initial message
6|    MOV R1, #msg1
7|    STR R1, .WriteString
8|//Computers Turn
9|select: LDR R2, .Random
10|    AND R2,R2, #3
11|    CMP R2, #0
12|    BEQ select
13|    CMP R2,R0
14|    BGT select
15|    BEQ select
16|cont: STR R2, .WriteUnsignedNum
17|    MOV R1, #msg4
18|    STR R1, .WriteString
19|    SUB R0, R0, R2
20|//Print remaining matchstick
21|    STR R0, .WriteUnsignedNum
22|    MOV R1, #msg1
23|    STR R1, .WriteString
24|//Check if computer wins
25|    cmp r0, #1
26|    BEQ computerWins
27|//Players turn

```

```
28|    MOV R1, #msg2
29|    STR R1, .WriteString
30|input: LDR R2, .InputNum
31|    CMP R2, #3
32|    BGT input
33|    CMP R2, #1
34|    BLT input
35|    CMP R2,R0
36|    BGT input
37|    SUB R0,R0,R2
38|    CMP R0,#1
39|    BEQ playerWins
40|    b Loop
41|playerWins: MOV R1, #msg3
42|    STR R1, .WriteString
43|    HALT
44|computerWins: MOV R1, #msg5
45|    STR R1, .WriteString
46|    HALT
47|msg1: .ASCIZ "remaining\n"
48|msg2: .ASCIZ "How many do you want to remove (1-3)?\n"
49|msg3: .ASCIZ "You Win!"
50|msg4: .ASCIZ "taken by computer\n"
51|msg5: .ASCIZ "Computer Wins!\n"
```