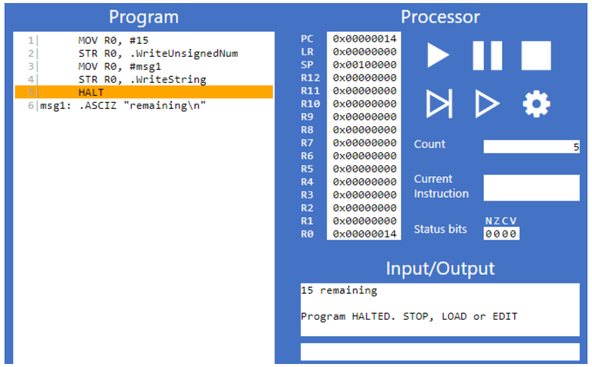
COS10004 - Computer Systems

Lab08 - 103234103 Son Nguyen

Lab 08 - Branching, Looping and Timers

8.1.1/

```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R0, #msg1
      STR R0, .WriteString
      HALT
msg1: .ASCIZ "remaining\n"
```



```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R0, #msg1
      STR R0, .WriteString
      HALT
msg1: .ASCIZ "remaining\n"
```

```
PC   0x00000014
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   0x00000000
R1   0x00000000
R0   0x00000014
```

Count 5

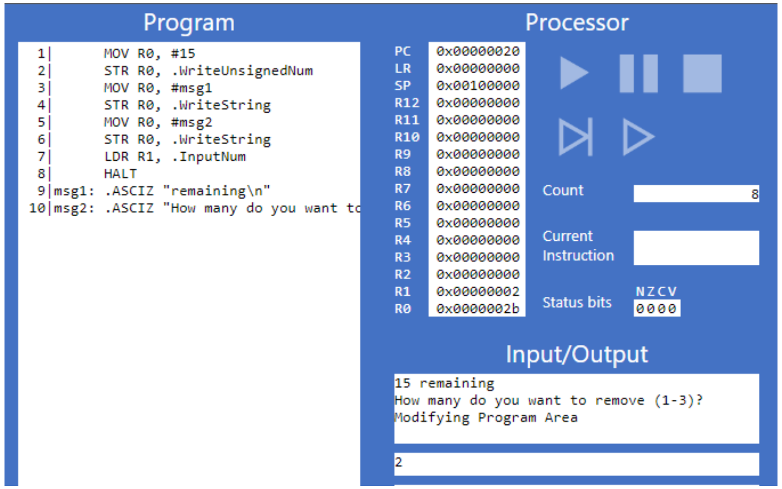
Current Instruction

Status bits NZCV 0000

```
15 remaining

Program HALTED. STOP, LOAD or EDIT
```

8.1.2/

```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R0, #msg1
      STR R0, .WriteString
      MOV R0, #msg2
      STR R0, .WriteString
      LDR R1, .InputNum
      HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want to remove (1-3)?"
```



```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R0, #msg1
      STR R0, .WriteString
      MOV R0, #msg2
      STR R0, .WriteString
      LDR R1, .InputNum
      HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want to
```

```
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   0x00000000
R2   0x00000000
R1   0x00000002
R0   0x0000002b
```

Count 8

Current Instruction

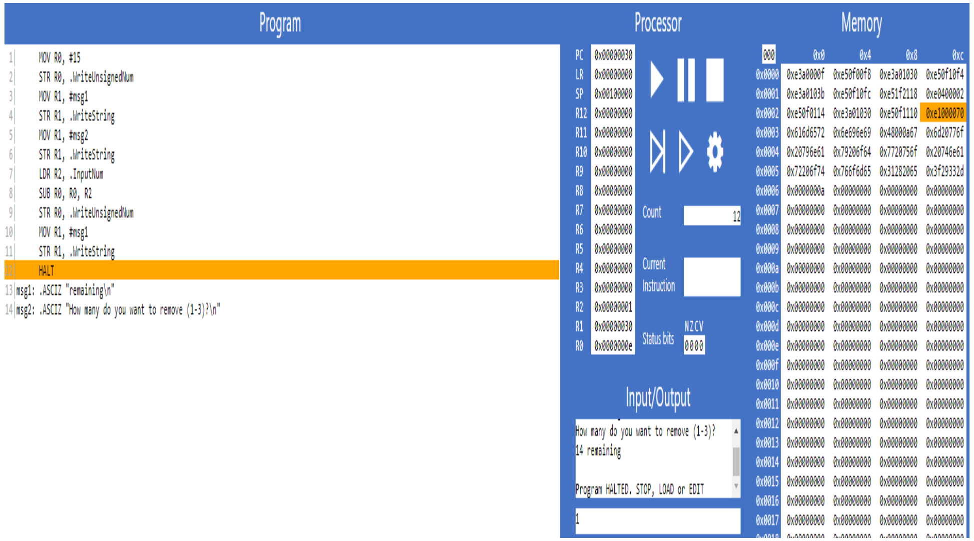
Status bits NZCV 0000

```
15 remaining
How many do you want to remove (1-3)?
Modifying Program Area
```

```
2
```

8.1.3/

```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
      MOV R1, #msg2
      STR R1, .WriteString
      LDR R2, .InputNum
      SUB R0, R0, R2
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
      HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want to remove (1-3)?\n"
```



```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
      MOV R1, #msg2
      STR R1, .WriteString
      LDR R2, .InputNum
      SUB R0, R0, R2
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
      HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want to remove (1-3)?\n"
```

```
PC   0x00000030
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   0x00000001
R1   0x00000030
R0   0x0000000e
```

Count 12

Status bits NZCV 0000

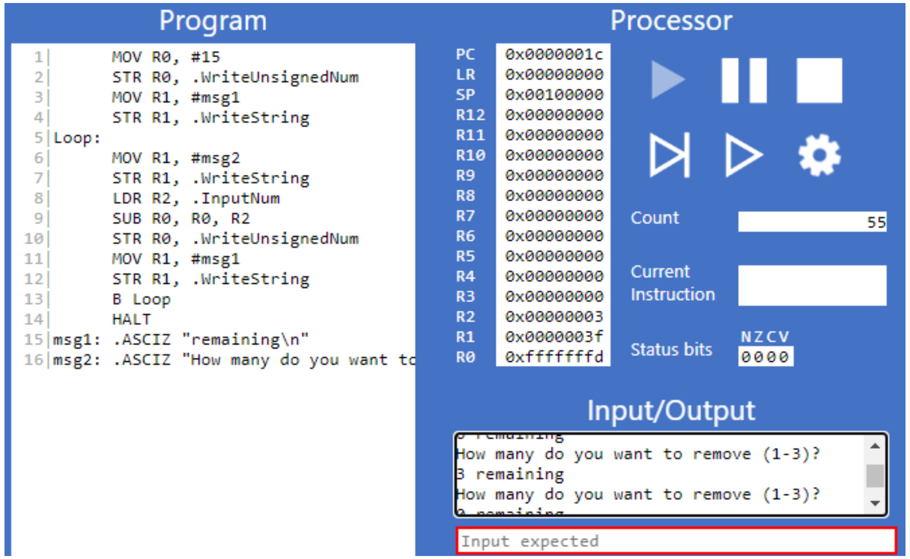
```
000         0x0         0x4         0x8         0xc
0x0000  0xe3a0000f  0xe50f00f8  0xe3a01030  0xe50f10f4
0x0001  0xe3a0103b  0xe50f10fc  0xe51f2118  0xe0400002
0x0002  0xe50f0114  0xe3a01030  0xe50f1110  0xe1000070
0x0003  0x616d6572  0x6e696e69  0x48000a67  0x6d20776f
0x0004  0x20796e61  0x79206f64  0x7720756f  0x20746e61
0x0005  0x72206f74  0x766f6d65  0x31282065  0x3f29332d
0x0006  0x0000000a  0x00000000  0x00000000  0x00000000
0x0007  0x00000000  0x00000000  0x00000000  0x00000000
0x0008  0x00000000  0x00000000  0x00000000  0x00000000
0x0009  0x00000000  0x00000000  0x00000000  0x00000000
0x000a  0x00000000  0x00000000  0x00000000  0x00000000
0x000b  0x00000000  0x00000000  0x00000000  0x00000000
0x000c  0x00000000  0x00000000  0x00000000  0x00000000
0x000d  0x00000000  0x00000000  0x00000000  0x00000000
0x000e  0x00000000  0x00000000  0x00000000  0x00000000
0x000f  0x00000000  0x00000000  0x00000000  0x00000000
0x0010  0x00000000  0x00000000  0x00000000  0x00000000
0x0011  0x00000000  0x00000000  0x00000000  0x00000000
0x0012  0x00000000  0x00000000  0x00000000  0x00000000
0x0013  0x00000000  0x00000000  0x00000000  0x00000000
0x0014  0x00000000  0x00000000  0x00000000  0x00000000
0x0015  0x00000000  0x00000000  0x00000000  0x00000000
0x0016  0x00000000  0x00000000  0x00000000  0x00000000
0x0017  0x00000000  0x00000000  0x00000000  0x00000000
```

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


Program HALTED. STOP, LOAD or EDIT
```

```
1
```

8.2.1/

```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
Loop:
      MOV R1, #msg2
      STR R1, .WriteString
      LDR R2, .InputNum
      SUB R0, R0, R2
       STR R0, .WriteUnsignedNum
       MOV R1, #msg1
       STR R1, .WriteString
       B Loop
       HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want to remove (1-3)?\n"
```



```
        MOV R0, #15
        STR R0, .WriteUnsignedNum
        MOV R1, #msg1
        STR R1, .WriteString
 Loop:
        MOV R1, #msg2
        STR R1, .WriteString
        LDR R2, .InputNum
        SUB R0, R0, R2
        STR R0, .WriteUnsignedNum
        MOV R1, #msg1
        STR R1, .WriteString
        B Loop
        HALT
 msg1: .ASCIZ "remaining\n"
 msg2: .ASCIZ "How many do you want to
```

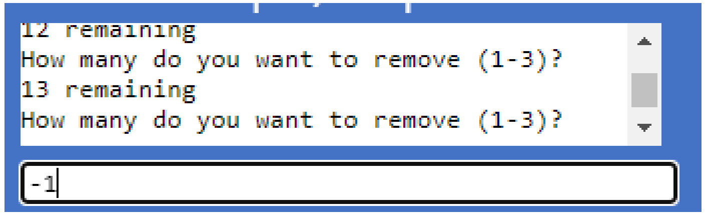
```
PC   0x0000001c
LR   0x00000000
SP   0x00100000
R12  0x00000000
R11  0x00000000
R10  0x00000000
R9   0x00000000
R8   0x00000000
R7   0x00000000
R6   0x00000000
R5   0x00000000
R4   0x00000000
R3   0x00000000
R2   0x00000003
R1   0x0000003f
R0   0xfffffffd

NZCV
0000
```

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

Input expected

8.2.1/ What happens if you enter a number that takes the number of matchsticks remaining beyond 0 (i.e., into negative values) ?

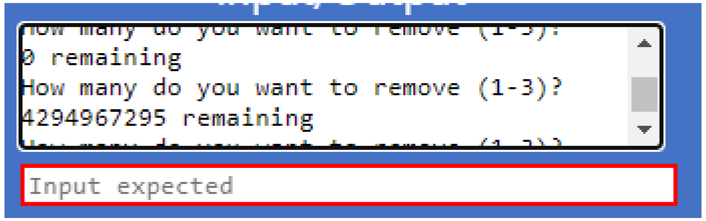
If the number input is a negative , let’s say -1 ; then the number of matchsticks would go up , eg. -1 into the program would bring it up



```
12 remaining
How many do you want to remove (1-3)?
13 remaining
How many do you want to remove (1-3)?
```

-1

While if you take all the values then it would go up to the maximum number in 32-bit unsigned integer because it cannot display



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

Input expected

negative

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)

0 < R2 < 4

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

Two assembly instructions could be used to create a branch that only occurs under the condition (BGT & BLT) , (BGT: Z clear , N and V being the same) , (BLT : where N and V are different).

(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 ?

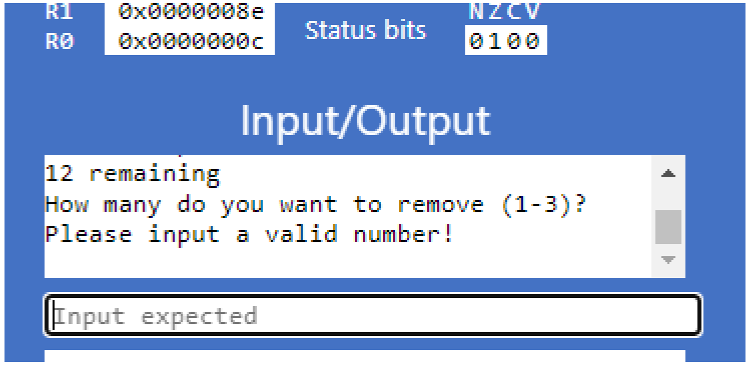
If the first condition is not met (R2 > 0) where R2 is negative then N = 1 ;

If the first condition is not met (R2 > 0) & 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 > 0) and (R2 = 4) then both Z and C = 1.

(d)



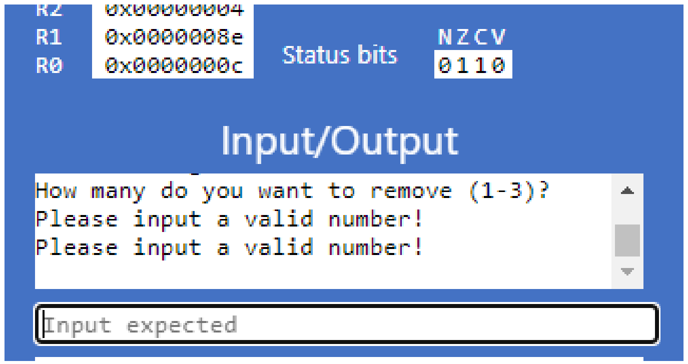
```
R1   0x0000008e
R0   0x0000000c
```

Status bits

NZCV
0100

Input/Output

```
12 remaining
How many do you want to remove (1-3)?
Please input a valid number!
```



```
R2   0x00000004
R1   0x0000008e
R0   0x0000000c
```

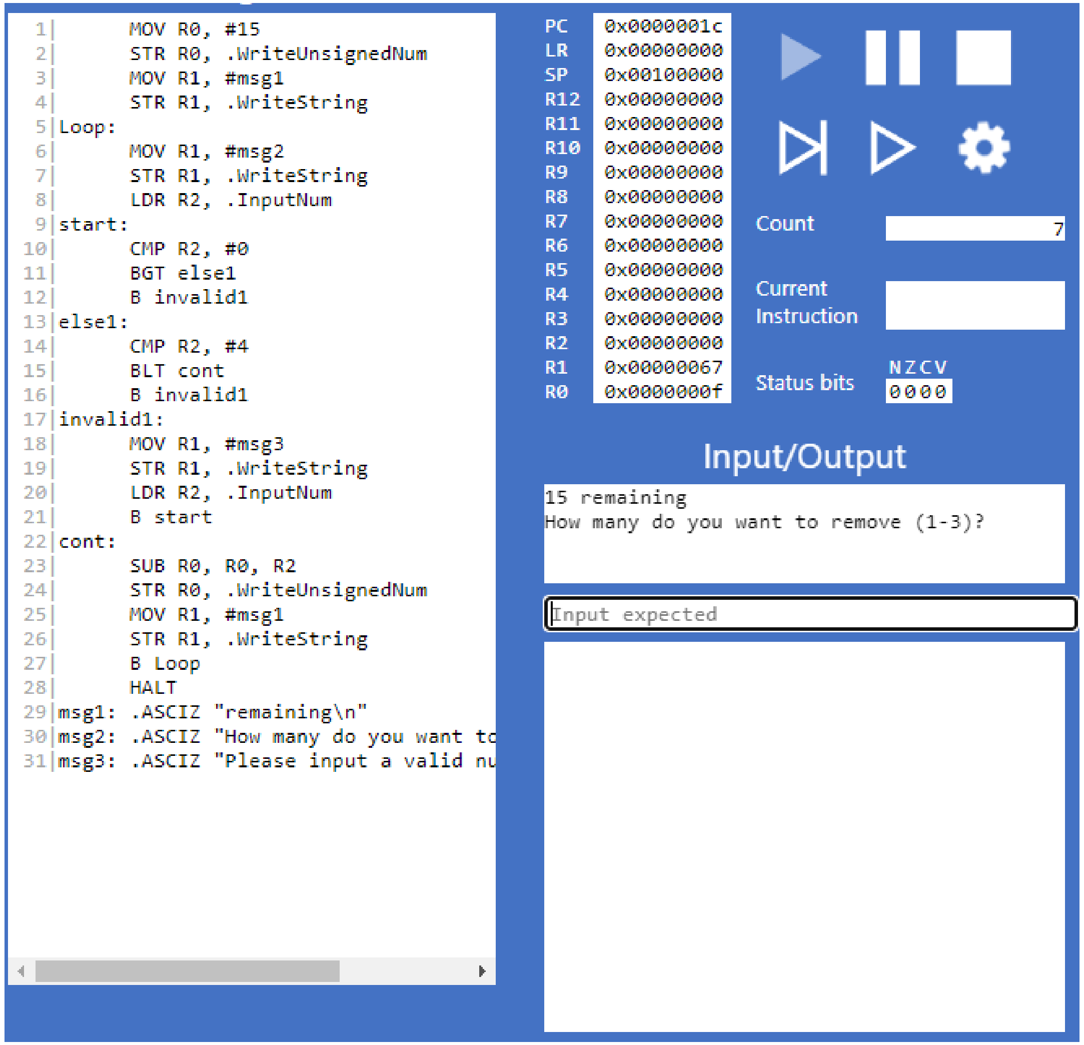
Status bits

NZCV
0110

Input/Output

```
How many do you want to remove (1-3)?
Please input a valid number!
Please input a valid number!
```

```
        MOV R0, #15
        STR R0, .WriteUnsignedNum
        MOV R1, #msg1
        STR R1, .WriteString
Loop:
        MOV R1, #msg2
        STR R1, .WriteString
        LDR R2, .InputNum
start:
        CMP R2, #0
        BGT else1
        B invalid1
else1:
        CMP R2, #4
        BLT cont
        B invalid1
invalid1:
        MOV R1, #msg3
        STR R1, .WriteString
        LDR R2, .InputNum
        B start
cont:
        SUB R0, R0, R2
        STR R0, .WriteUnsignedNum
        MOV R1, #msg1
        STR R1, .WriteString
        B Loop
        HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want to remove (1-3)?\n"
msg3: .ASCIZ "Please input a valid number!\n"
```



```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
Loop:
      MOV R1, #msg2
      STR R1, .WriteString
      LDR R2, .InputNum
start:
      CMP R2, #0
      BGT else1
      B invalid1
else1:
      CMP R2, #4
      BLT cont
      B invalid1
invalid1:
      MOV R1, #msg3
      STR R1, .WriteString
      LDR R2, .InputNum
      B start
cont:
      SUB R0, R0, R2
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
      B Loop
      HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want tc
msg3: .ASCIZ "Please input a valid nu
```

```
PC   0x0000001c
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   0x00000000
R1   0x00000067
R0   0x0000000f
```

Count 7

Current Instruction

Status bits NZCV 0000

Input/Output

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

Input expected

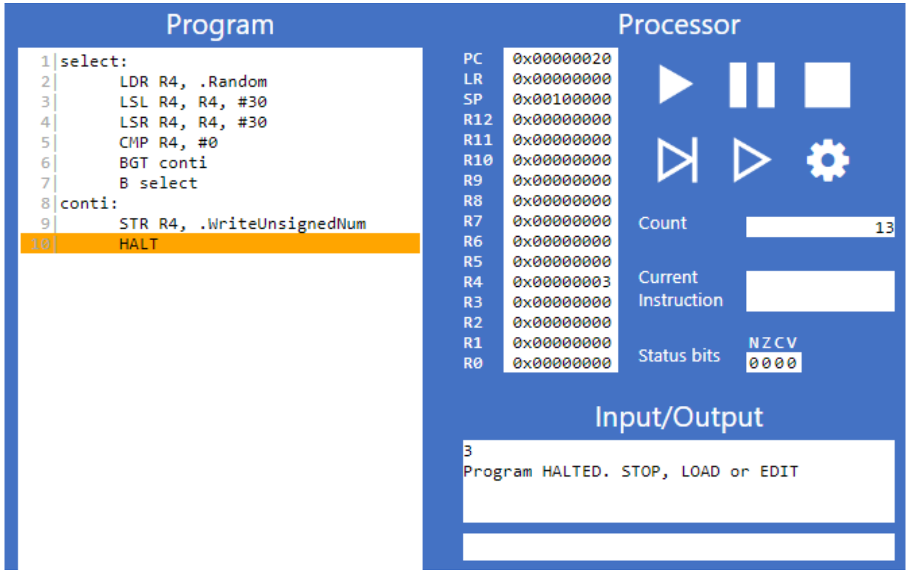
8.3.1/

(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.

```
LSL R4, R4, #30
LSR R4, R4, #30
```

(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.

```
select:
     LDR R4, .Random
     LSL R4, R4, #30
     LSR R4, R4, #30
     CMP R4, #0
     BGT conti
     B select
conti:
     STR R4, .WriteUnsignedNum
      HALT
```



```
select:
        LDR R4, .Random
        LSL R4, R4, #30
        LSR R4, R4, #30
        CMP R4, #0
        BGT conti
        B select
conti:
        STR R4, .WriteUnsignedNum
        HALT
```

```
PC   0x00000020
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   0x00000000
```

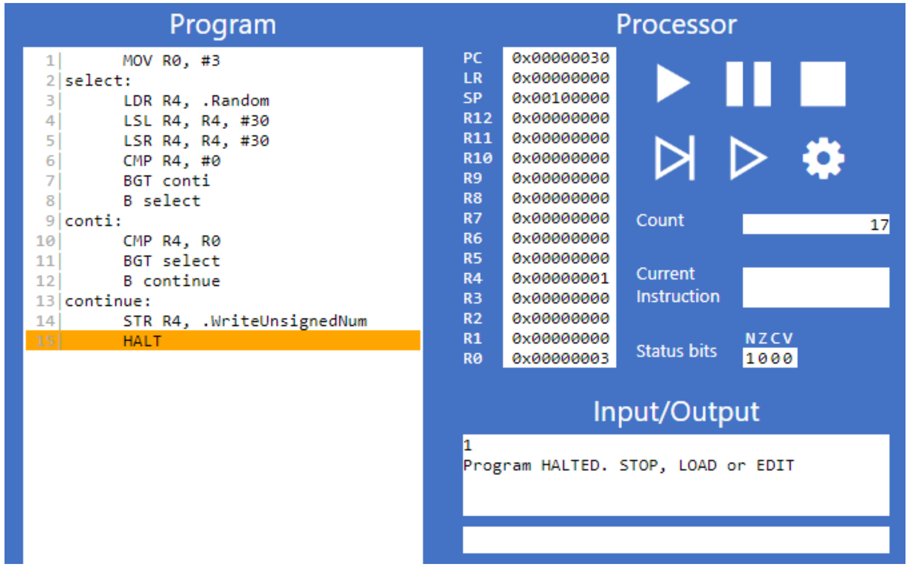
Count 13

Status bits NZCV 0000

```
3
Program HALTED. STOP, LOAD or EDIT
```

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
```



```
      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
```

```
PC   0x00000030
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   0x00000003
```

Count 17

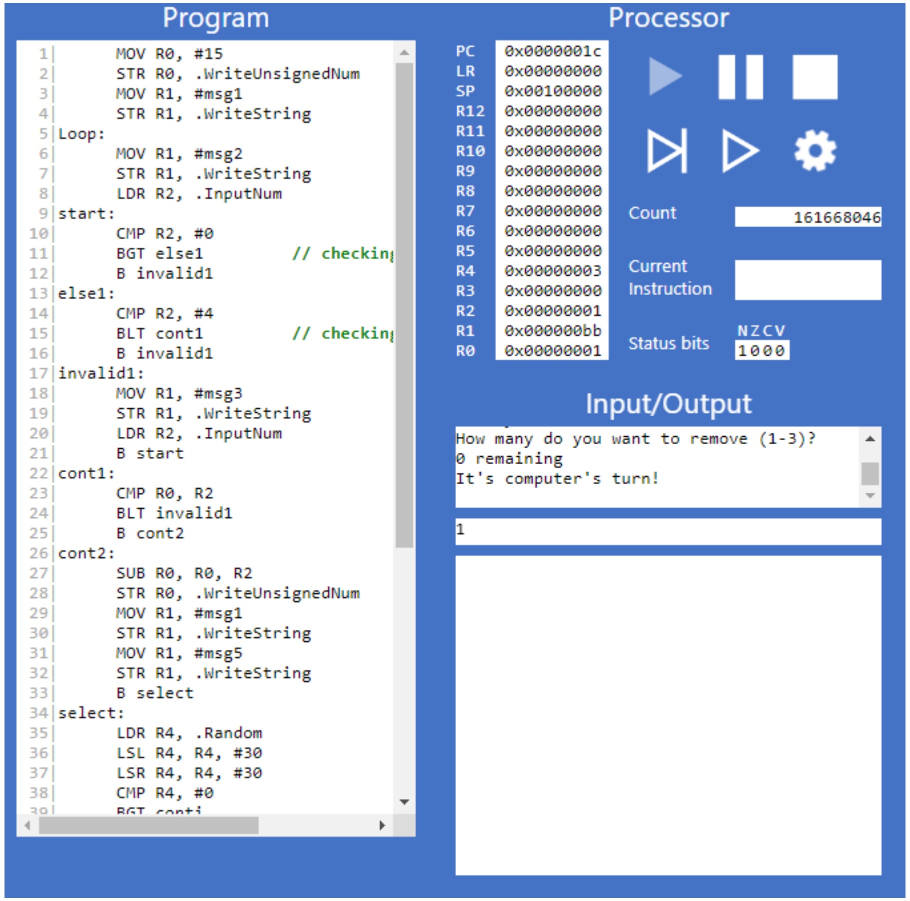
Current Instruction

Status bits NZCV 1000

```
1
Program HALTED. STOP, LOAD or EDIT
```

8.4.1/

```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
Loop:
      MOV R1, #msg2
      STR R1, .WriteString
      LDR R2, .InputNum
start:
      CMP R2, #0
      BGT else1         // checking if R2 > 0
      B invalid1
else1:
      CMP R2, #4
      BLT cont1         // checking if R2 < 4
      B invalid1
invalid1:
      MOV R1, #msg3
      STR R1, .WriteString
      LDR R2, .InputNum
      B start
cont1:
      CMP R0, R2
      BLT invalid1
      B cont2
cont2:
      SUB R0, R0, R2
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
      MOV R1, #msg5
      STR R1, .WriteString
      B select
select:
```



```
      MOV R0, #15
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
Loop:
      MOV R1, #msg2
      STR R1, .WriteString
      LDR R2, .InputNum
start:
      CMP R2, #0
      BGT else1         // checkin
      B invalid1
else1:
      CMP R2, #4
      BLT cont1         // checkin
      B invalid1
invalid1:
      MOV R1, #msg3
      STR R1, .WriteString
      LDR R2, .InputNum
      B start
cont1:
      CMP R0, R2
      BLT invalid1
      B cont2
cont2:
      SUB R0, R0, R2
      STR R0, .WriteUnsignedNum
      MOV R1, #msg1
      STR R1, .WriteString
      MOV R1, #msg5
      STR R1, .WriteString
      B select
select:
      LDR R4, .Random
      LSL R4, R4, #30
      LSR R4, R4, #30
      CMP R4, #0
      BGT conti
```

PC	0x0000001c
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	0x00000001
R1	0x000000bb
R0	0x00000001

Count 161668046

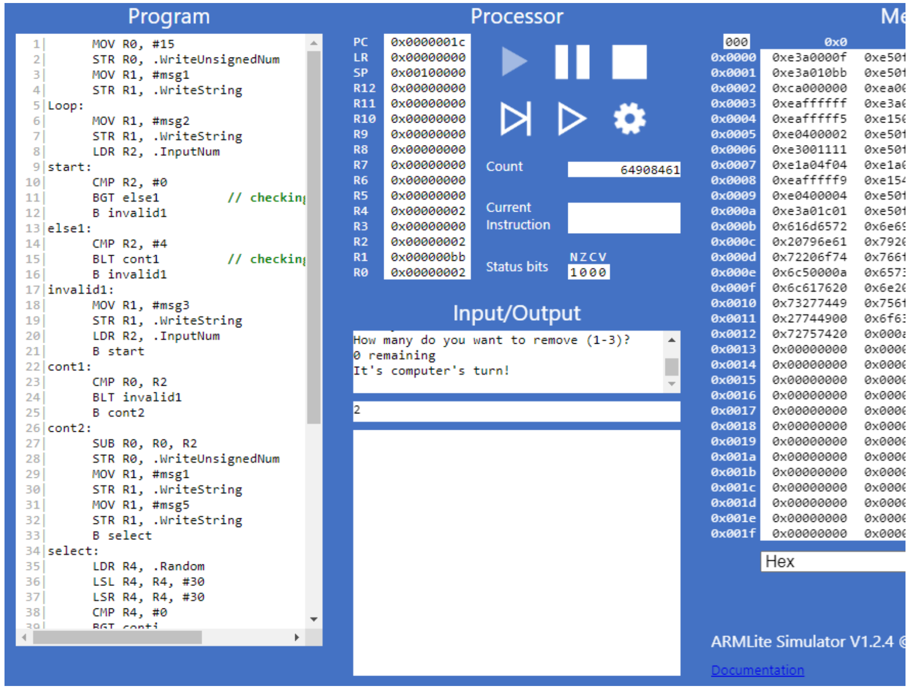
Current Instruction

Status bits NZCV 1000

```
How many do you want to remove (1-3)?
0 remaining
It's computer's turn!
```

1

```
        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:
        SUB R0, R0, R4
        STR R0, .WriteUnsignedNum
        MOV R1, #msg1
        STR R1, .WriteString
        MOV R1, #msg4
        STR R1, .WriteString
        B Loop
        HALT
msg1: .ASCIZ "remaining\n"
msg2: .ASCIZ "How many do you want to remove (1-3)?\n"
msg3: .ASCIZ "Please input a valid number!\n"
msg4: .ASCIZ "It's your turn!\n"
msg5: .ASCIZ "It's computer's turn!\n"
```



```
        MOV R0, #15
        STR R0, .WriteUnsignedNum
        MOV R1, #msg1
        STR R1, .WriteString
Loop:
        MOV R1, #msg2
        STR R1, .WriteString
        LDR R2, .InputNum
start:
        CMP R2, #0
        BGT else1           // checkin
        B invalid1
else1:
        CMP R2, #4
        BLT cont1           // checkin
        B invalid1
invalid1:
        MOV R1, #msg3
        STR R1, .WriteString
        LDR R2, .InputNum
        B start
cont1:
        CMP R0, R2
        BLT invalid1
        B cont2
cont2:
        SUB R0, R0, R2
        STR R0, .WriteUnsignedNum
        MOV R1, #msg1
        STR R1, .WriteString
        MOV R1, #msg5
        STR R1, .WriteString
        B select
select:
        LDR R4, .Random
        LSL R4, R4, #30
        LSR R4, R4, #30
        CMP R4, #0
        BGT conti
```

```
PC   0x0000001c
LR   0x00000000
SP   0x00100000
R12  0x00000000
R11  0x00000000
R10  0x00000000
R9   0x00000000
R8   0x00000000
R7   0x00000000
R6   0x00000000
R5   0x00000000
R4   0x00000002
R3   0x00000000
R2   0x00000002
R1   0x000000bb
R0   0x00000002
```

Count 64908461

Current Instruction

Status bits NZCV 1000

```
How many do you want to remove (1-3)?
0 remaining
It's computer's turn!
```

```
2
```

```
000     0x0
0x0000  0xe3a0000f  0xe50
0x0001  0xe3a010bb  0xe50
0x0002  0xca000000  0xea0
0x0003  0xeaffffff  0xe3a
0x0004  0xeafffff5  0xe15
0x0005  0xe0400002  0xe50
0x0006  0xe3001111  0xe50
0x0007  0xe1a04f04  0xe1a
0x0008  0xeafffff9  0xe15
0x0009  0xe0400004  0xe50
0x000a  0xe3a01c01  0xe50
0x000b  0x616d6572  0x6e6
0x000c  0x20796e61  0x792
0x000d  0x72206f74  0x766
0x000e  0x6c50000a  0x657
0x000f  0x6c617620  0x6e2
0x0010  0x73277449  0x756
0x0011  0x27744900  0x6f6
0x0012  0x72757420  0x000
0x0013  0x00000000  0x000
0x0014  0x00000000  0x000
0x0015  0x00000000  0x000
0x0016  0x00000000  0x000
0x0017  0x00000000  0x000
0x0018  0x00000000  0x000
0x0019  0x00000000  0x000
0x001a  0x00000000  0x000
0x001b  0x00000000  0x000
0x001c  0x00000000  0x000
0x001d  0x00000000  0x000
0x001e  0x00000000  0x000
0x001f  0x00000000  0x000
```