EECS2021

Lab C

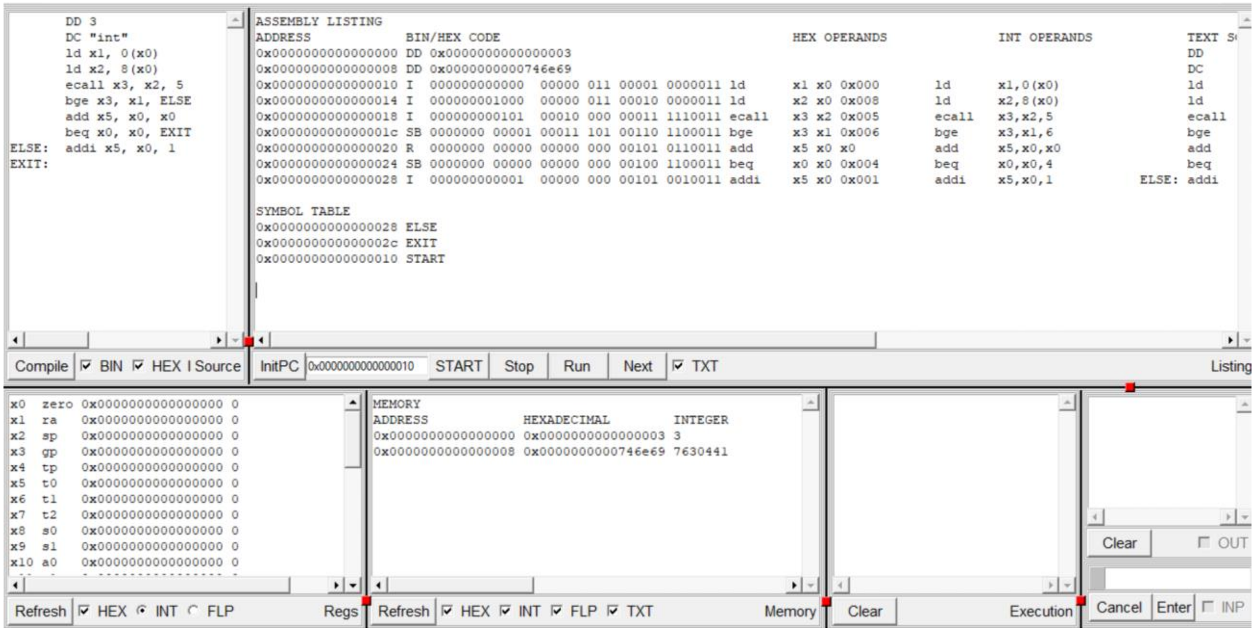
Bottom text

jonathan lai
[Date]

Program 1:

```
        DD 3
        DC "int"
        ld x1, 0(x0)
        ld x2, 8(x0)
        ecall x3, x2, 5
        bge x3, x1, ELSE
        add x5, x0, x0
        beq x0, x0, EXIT
ELSE:   addi x5, x0, 1
EXIT:
```

After compiling



```
       DD 3
       DC "int"
       ld x1, 0(x0)
       ld x2, 8(x0)
       ecall x3, x2, 5
       bge x3, x1, ELSE
       add x5, x0, x0
       beq x0, x0, EXIT
ELSE:  addi x5, x0, 1
EXIT:
```

```
ASSEMBLY LISTING
ADDRESS            BIN/HEX CODE                                      HEX OPERANDS          INT OPERANDS         TEXT S
0x0000000000000000 DD 0x0000000000000003                                                                        DD
0x0000000000000008 DD 0x00000000000746e69                                                                       DC
0x0000000000000010 I  000000000000  00000 011 00001 0000011 ld       x1 x0 0x000    ld     x1,0(x0)             ld
0x0000000000000014 I  000000001000  00000 011 00010 0000011 ld       x2 x0 0x008    ld     x2,8(x0)             ld
0x0000000000000018 I  000000000101  00010 000 00011 1110011 ecall    x3 x2 0x005    ecall  x3,x2,5              ecall
0x000000000000001c SB 0000000 00001 00011 101 00110 1100011 bge      x3 x1 0x006    bge    x3,x1,6              bge
0x0000000000000020 R  0000000 00000 00000 000 00101 0110011 add      x5 x0 x0       add    x5,x0,x0             add
0x0000000000000024 SB 0000000 00000 00000 000 00100 1100011 beq      x0 x0 0x004    beq    x0,x0,4              beq
0x0000000000000028 I  000000000001  00000 000 00101 0010011 addi     x5 x0 0x001    addi   x5,x0,1        ELSE: addi

SYMBOL TABLE
0x0000000000000028 ELSE
0x000000000000002c EXIT
0x0000000000000010 START
```

Compile ☑ BIN ☑ HEX I Source
InitPC 0x0000000000000010 START Stop Run Next ☑ TXT
Listing

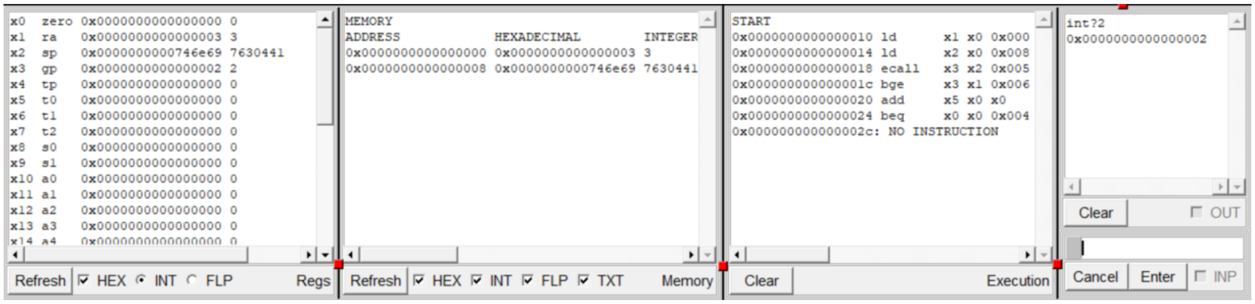
```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000000 0
x2  sp   0x0000000000000000 0
x3  gp   0x0000000000000000 0
x4  tp   0x0000000000000000 0
x5  t0   0x0000000000000000 0
x6  t1   0x0000000000000000 0
x7  t2   0x0000000000000000 0
x8  s0   0x0000000000000000 0
x9  s1   0x0000000000000000 0
x10 a0   0x0000000000000000 0
```

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x000000000746e69 7630441
```

Refresh ☑ HEX ● INT ○ FLP Regs
Refresh ☑ HEX ☑ INT ☑ FLP ☑ TXT Memory
Clear Execution
Clear □ OUT
Cancel Enter □ INP

After running:

A>B:



```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000003 3
x2  sp   0x000000000746e69 7630441
x3  gp   0x0000000000000002 2
x4  tp   0x0000000000000000 0
x5  t0   0x0000000000000000 0
x6  t1   0x0000000000000000 0
x7  t2   0x0000000000000000 0
x8  s0   0x0000000000000000 0
x9  s1   0x0000000000000000 0
x10 a0   0x0000000000000000 0
x11 a1   0x0000000000000000 0
x12 a2   0x0000000000000000 0
x13 a3   0x0000000000000000 0
x14 a4   0x0000000000000000 0
```

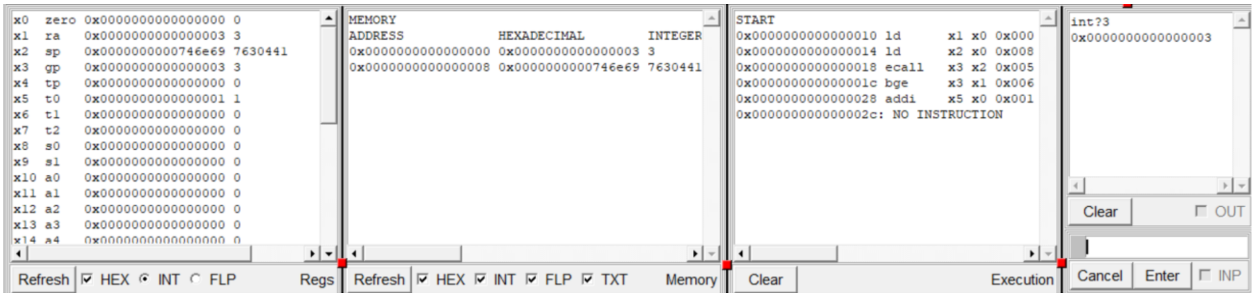
```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x000000000746e69 7630441
```

```
START
0x0000000000000010 ld      x1 x0 0x000
0x0000000000000014 ld      x2 x0 0x008
0x0000000000000018 ecall   x3 x2 0x005
0x000000000000001c bge     x3 x1 0x006
0x0000000000000020 add     x5 x0 x0
0x0000000000000024 beq     x0 x0 0x004
0x000000000000002c: NO INSTRUCTION
```

```
int?2
0x0000000000000002
```

Refresh ☑ HEX ● INT ○ FLP Regs
Refresh ☑ HEX ☑ INT ☑ FLP ☑ TXT Memory
Clear Execution
Clear □ OUT
Cancel Enter □ INP

A == B:



```
x0   zero 0x0000000000000000 0
x1   ra   0x0000000000000003 3
x2   sp   0x0000000000746e69 7630441
x3   gp   0x0000000000000003 3
x4   tp   0x0000000000000000 0
x5   t0   0x0000000000000001 1
x6   t1   0x0000000000000000 0
x7   t2   0x0000000000000000 0
x8   s0   0x0000000000000000 0
x9   s1   0x0000000000000000 0
x10  a0   0x0000000000000000 0
x11  a1   0x0000000000000000 0
x12  a2   0x0000000000000000 0
x13  a3   0x0000000000000000 0
x14  a4   0x0000000000000000 0
```

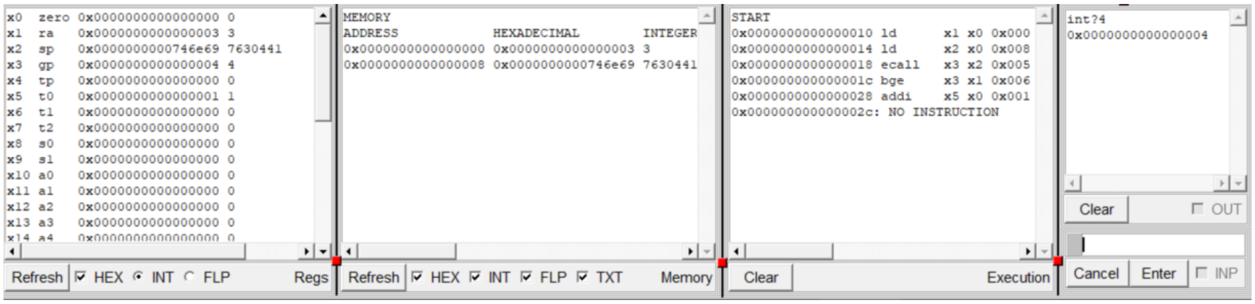
```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x0000000000746e69 7630441
```

```
START
0x0000000000000010 ld       x1 x0 0x000
0x0000000000000014 ld       x2 x0 0x008
0x0000000000000018 ecall    x3 x2 0x005
0x000000000000001c bge      x3 x1 0x006
0x0000000000000028 addi     x5 x0 0x001
0x000000000000002c: NO INSTRUCTION
```

```
int?3
0x0000000000000003
```

Refresh ☑ HEX ● INT ○ FLP Regs | Refresh ☑ HEX ☑ INT ☑ FLP ☑ TXT Memory | Clear Execution | Clear □ OUT | Cancel Enter □ INP

A < B:



```
x0   zero 0x0000000000000000 0
x1   ra   0x0000000000000003 3
x2   sp   0x0000000000746e69 7630441
x3   gp   0x0000000000000004 4
x4   tp   0x0000000000000000 0
x5   t0   0x0000000000000001 1
x6   t1   0x0000000000000000 0
x7   t2   0x0000000000000000 0
x8   s0   0x0000000000000000 0
x9   s1   0x0000000000000000 0
x10  a0   0x0000000000000000 0
x11  a1   0x0000000000000000 0
x12  a2   0x0000000000000000 0
x13  a3   0x0000000000000000 0
x14  a4   0x0000000000000000 0
```

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x0000000000746e69 7630441
```

```
START
0x0000000000000010 ld       x1 x0 0x000
0x0000000000000014 ld       x2 x0 0x008
0x0000000000000018 ecall    x3 x2 0x005
0x000000000000001c bge      x3 x1 0x006
0x0000000000000028 addi     x5 x0 0x001
0x000000000000002c: NO INSTRUCTION
```

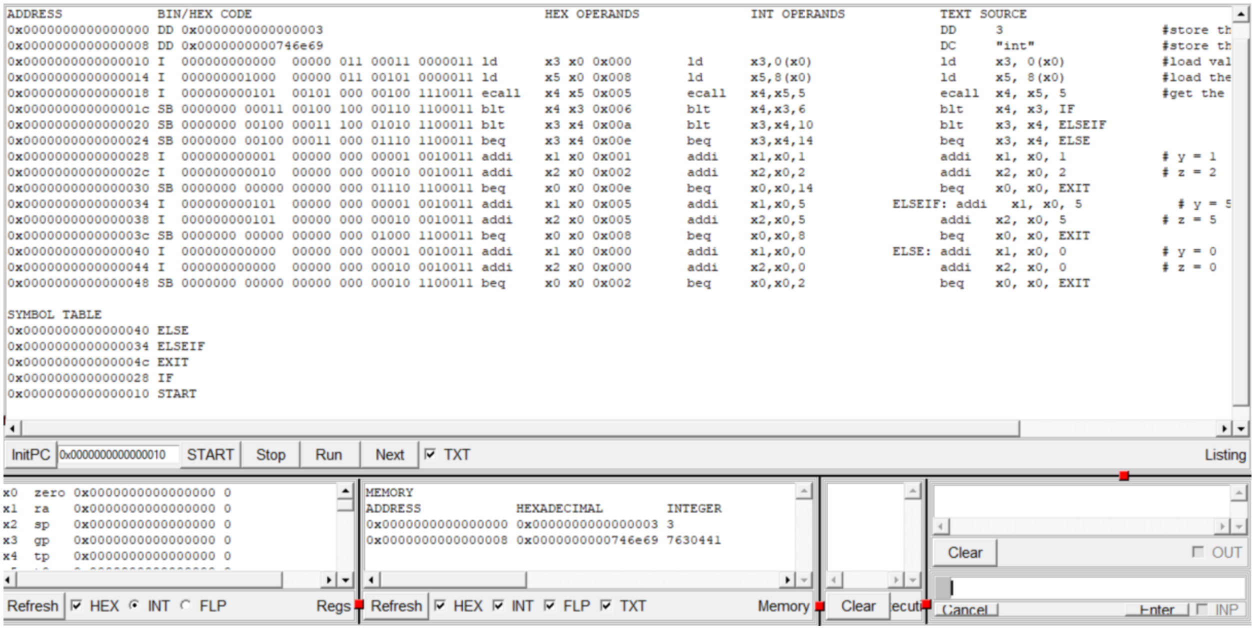
```
int?4
0x0000000000000004
```

Refresh ☑ HEX ● INT ○ FLP Regs | Refresh ☑ HEX ☑ INT ☑ FLP ☑ TXT Memory | Clear Execution | Clear □ OUT | Cancel Enter □ INP

Program 2:

```
DD 3                        #store the value of A
DC "int"                    #store the prompt for the input inte
ld x3, 0(x0)                #load value of A to x3
ld x5, 8(x0)                #load the prompt
ecall x4, x5, 5             #get the input for B

blt x4, x3, IF
blt x3, x4, ELSEIF
beq x3, x4, ELSE


IF:
addi x1, x0, 1 # y = 1
addi x2, x0, 2 # z = 2
beq x0, x0, EXIT

ELSEIF: addi x1, x0, 5 # y = 5
        addi x2, x0, 5 # z = 5
        beq x0, x0, EXIT
ELSE: addi x1, x0, 0 # y = 0
      addi x2, x0, 0 # z = 0
      beq x0, x0, EXIT
EXIT:
```

After compiling:



```
ADDRESS            BIN/HEX CODE                                           HEX OPERANDS         INT OPERANDS         TEXT SOURCE
0x0000000000000000 DD 0x0000000000000003                                                                              DD     3                   #store th
0x0000000000000008 DD 0x0000000000746e69                                                                              DC     "int"               #store th
0x0000000000000010 I  000000000000  00000 011 00011 0000011 ld     x3 x0 0x000    ld      x3,0(x0)            ld     x3, 0(x0)           #load val
0x0000000000000014 I  000000001000  00000 011 00101 0000011 ld     x5 x0 0x008    ld      x5,8(x0)            ld     x5, 8(x0)           #load the
0x0000000000000018 I  000000000101  00101 000 00100 1110011 ecall  x4 x5 0x005    ecall   x4,x5,5             ecall  x4, x5, 5           #get the
0x000000000000001c SB 0000000 00011 00100 100 00110 1100011 blt    x4 x3 0x006    blt     x4,x3,6             blt    x4, x3, IF
0x0000000000000020 SB 0000000 00100 00011 100 01010 1100011 blt    x3 x4 0x00a    blt     x3,x4,10            blt    x3, x4, ELSEIF
0x0000000000000024 SB 0000000 00100 00011 000 01110 1100011 beq    x3 x4 0x00e    beq     x3,x4,14            beq    x3, x4, ELSE
0x0000000000000028 I  000000000001  00000 000 00001 0010011 addi   x1 x0 0x001    addi    x1,x0,1             addi   x1, x0, 1           # y = 1
0x000000000000002c I  000000000010  00000 000 00010 0010011 addi   x2 x0 0x002    addi    x2,x0,2             addi   x2, x0, 2           # z = 2
0x0000000000000030 SB 0000000 00000 00000 000 01110 1100011 beq    x0 x0 0x00e    beq     x0,x0,14            beq    x0, x0, EXIT
0x0000000000000034 I  000000000101  00000 000 00001 0010011 addi   x1 x0 0x005    addi    x1,x0,5      ELSEIF: addi     x1, x0, 5            # y = 5
0x0000000000000038 I  000000000101  00000 000 00010 0010011 addi   x2 x0 0x005    addi    x2,x0,5             addi   x2, x0, 5           # z = 5
0x000000000000003c SB 0000000 00000 00000 000 01000 1100011 beq    x0 x0 0x008    beq     x0,x0,8             beq    x0, x0, EXIT
0x0000000000000040 I  000000000000  00000 000 00001 0010011 addi   x1 x0 0x000    addi    x1,x0,0      ELSE:  addi   x1, x0, 0           # y = 0
0x0000000000000044 I  000000000000  00000 000 00010 0010011 addi   x2 x0 0x000    addi    x2,x0,0             addi   x2, x0, 0           # z = 0
0x0000000000000048 SB 0000000 00000 00000 000 00010 1100011 beq    x0 x0 0x002    beq     x0,x0,2             beq    x0, x0, EXIT

SYMBOL TABLE
0x0000000000000040 ELSE
0x0000000000000034 ELSEIF
0x000000000000004c EXIT
0x0000000000000028 IF
0x0000000000000010 START
```

InitPC 0x0000000000000010

```
x0   zero  0x0000000000000000 0
x1   ra    0x0000000000000000 0
x2   sp    0x0000000000000000 0
x3   gp    0x0000000000000000 0
x4   tp    0x0000000000000000 0
```

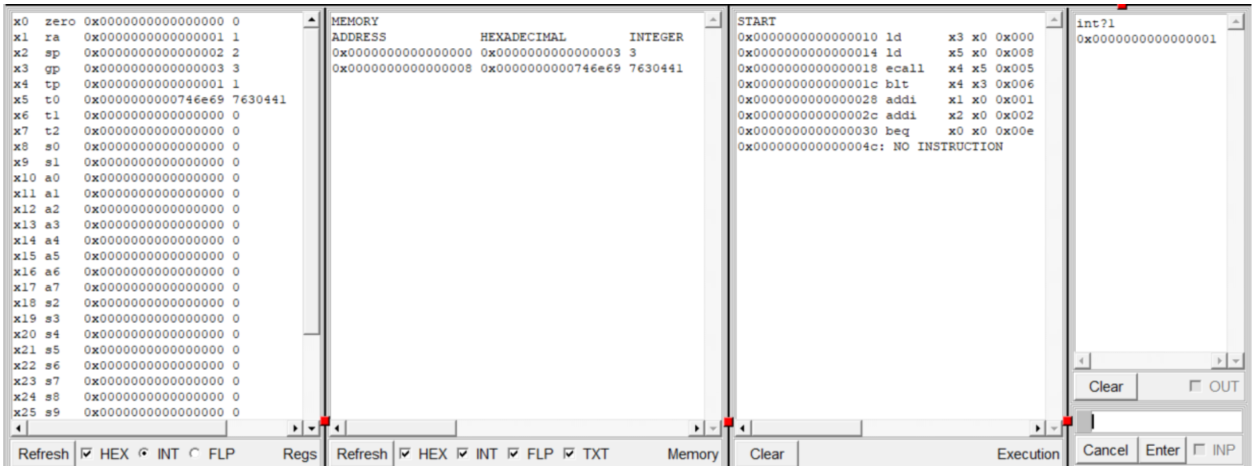
☑ HEX ● INT ○ FLP

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x0000000000746e69 7630441
```

☑ HEX ☑ INT ☑ FLP ☑ TXT

After running:

A>B:



```
x0   zero  0x0000000000000000 0
x1   ra    0x0000000000000001 1
x2   sp    0x0000000000000002 2
x3   gp    0x0000000000000003 3
x4   tp    0x0000000000000001 1
x5   t0    0x0000000000746e69 7630441
x6   t1    0x0000000000000000 0
x7   t2    0x0000000000000000 0
x8   s0    0x0000000000000000 0
x9   s1    0x0000000000000000 0
x10  a0    0x0000000000000000 0
x11  a1    0x0000000000000000 0
x12  a2    0x0000000000000000 0
x13  a3    0x0000000000000000 0
x14  a4    0x0000000000000000 0
x15  a5    0x0000000000000000 0
x16  a6    0x0000000000000000 0
x17  a7    0x0000000000000000 0
x18  s2    0x0000000000000000 0
x19  s3    0x0000000000000000 0
x20  s4    0x0000000000000000 0
x21  s5    0x0000000000000000 0
x22  s6    0x0000000000000000 0
x23  s7    0x0000000000000000 0
x24  s8    0x0000000000000000 0
x25  s9    0x0000000000000000 0
```

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x0000000000746e69 7630441
```

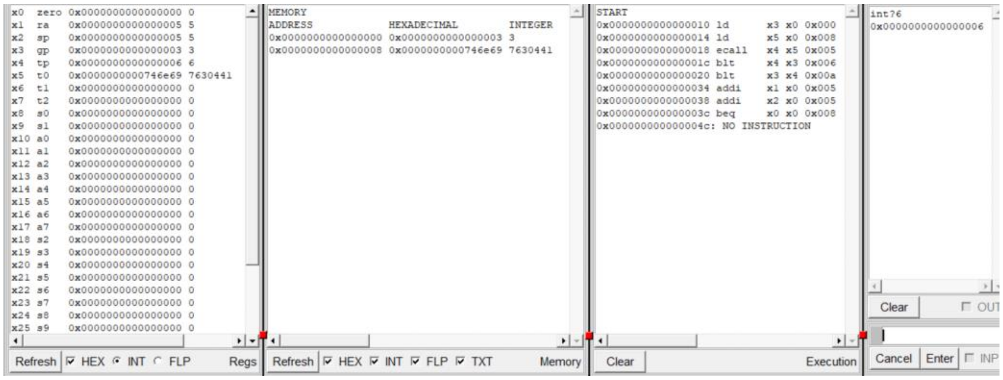
```
START
0x0000000000000010 ld       x3 x0 0x000
0x0000000000000014 ld       x5 x0 0x008
0x0000000000000018 ecall    x4 x5 0x005
0x000000000000001c blt      x4 x3 0x006
0x0000000000000028 addi     x1 x0 0x001
0x000000000000002c addi     x2 x0 0x002
0x0000000000000030 beq      x0 x0 0x00e
0x000000000000004c: NO INSTRUCTION
```

```
int?1
0x0000000000000001
```

☑ HEX ● INT ○ FLP

☑ HEX ☑ INT ☑ FLP ☑ TXT

A<B:



```
x0   zero  0x0000000000000000 0
x1   ra    0x0000000000000005 5
x2   sp    0x0000000000000005 5
x3   gp    0x0000000000000003 3
x4   tp    0x0000000000000006 6
x5   t0    0x0000000000746e69 7630441
x6   t1    0x0000000000000000 0
x7   t2    0x0000000000000000 0
x8   s0    0x0000000000000000 0
x9   s1    0x0000000000000000 0
x10  a0    0x0000000000000000 0
x11  a1    0x0000000000000000 0
x12  a2    0x0000000000000000 0
x13  a3    0x0000000000000000 0
x14  a4    0x0000000000000000 0
x15  a5    0x0000000000000000 0
x16  a6    0x0000000000000000 0
x17  a7    0x0000000000000000 0
x18  s2    0x0000000000000000 0
x19  s3    0x0000000000000000 0
x20  s4    0x0000000000000000 0
x21  s5    0x0000000000000000 0
x22  s6    0x0000000000000000 0
x23  s7    0x0000000000000000 0
x24  s8    0x0000000000000000 0
x25  s9    0x0000000000000000 0
```

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x0000000000746e69 7630441
```

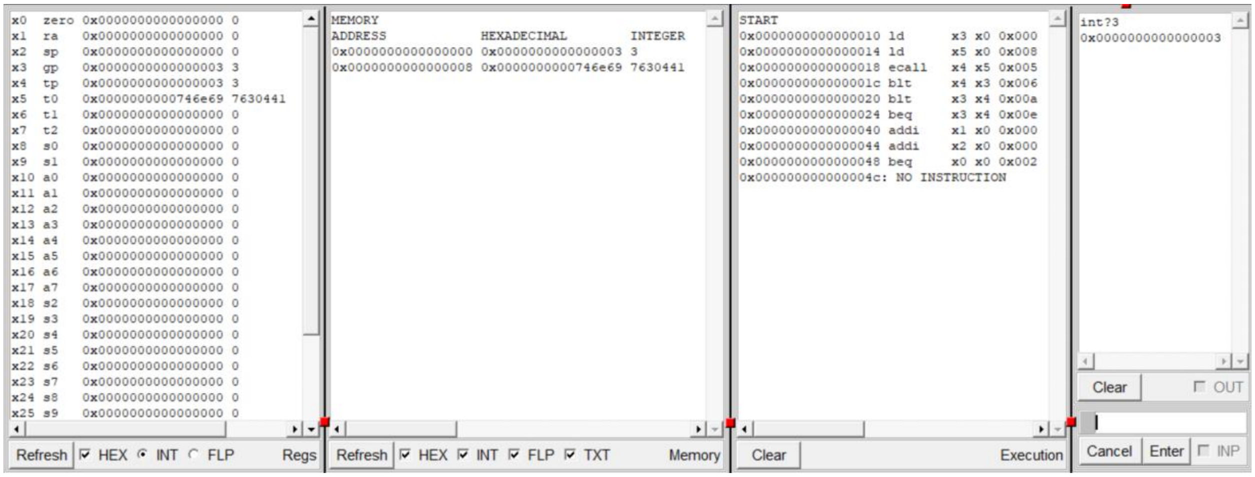
```
START
0x0000000000000010 ld       x3 x0 0x000
0x0000000000000014 ld       x5 x0 0x008
0x0000000000000018 ecall    x4 x5 0x005
0x000000000000001c blt      x4 x3 0x006
0x0000000000000020 blt      x3 x4 0x00a
0x0000000000000034 addi     x1 x0 0x005
0x0000000000000038 addi     x2 x0 0x005
0x000000000000003c beq      x0 x0 0x008
0x000000000000004c: NO INSTRUCTION
```

```
int?6
0x0000000000000006
```

☑ HEX ● INT ○ FLP

☑ HEX ☑ INT ☑ FLP ☑ TXT

A==B:



```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000000 0
x2  sp   0x0000000000000000 0
x3  gp   0x0000000000000003 3
x4  tp   0x0000000000000003 3
x5  t0   0x0000000000746e69 7630441
x6  t1   0x0000000000000000 0
x7  t2   0x0000000000000000 0
x8  s0   0x0000000000000000 0
x9  s1   0x0000000000000000 0
x10 a0   0x0000000000000000 0
x11 a1   0x0000000000000000 0
x12 a2   0x0000000000000000 0
x13 a3   0x0000000000000000 0
x14 a4   0x0000000000000000 0
x15 a5   0x0000000000000000 0
x16 a6   0x0000000000000000 0
x17 a7   0x0000000000000000 0
x18 s2   0x0000000000000000 0
x19 s3   0x0000000000000000 0
x20 s4   0x0000000000000000 0
x21 s5   0x0000000000000000 0
x22 s6   0x0000000000000000 0
x23 s7   0x0000000000000000 0
x24 s8   0x0000000000000000 0
x25 s9   0x0000000000000000 0
```

Refresh ☑ HEX ● INT ○ FLP Regs

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0000000000000003 3
0x0000000000000008 0x0000000000746e69 7630441
```

Refresh ☑ HEX ☑ INT ☑ FLP ☑ TXT Memory

```
START
0x0000000000000010 ld       x3 x0 0x000
0x0000000000000014 ld       x5 x0 0x008
0x0000000000000018 ecall    x4 x5 0x005
0x000000000000001c blt      x4 x3 0x006
0x0000000000000020 blt      x3 x4 0x00a
0x0000000000000024 beq      x3 x4 0x00e
0x0000000000000040 addi     x1 x0 0x000
0x0000000000000044 addi     x2 x0 0x000
0x0000000000000048 beq      x0 x0 0x002
0x000000000000004c: NO INSTRUCTION
```

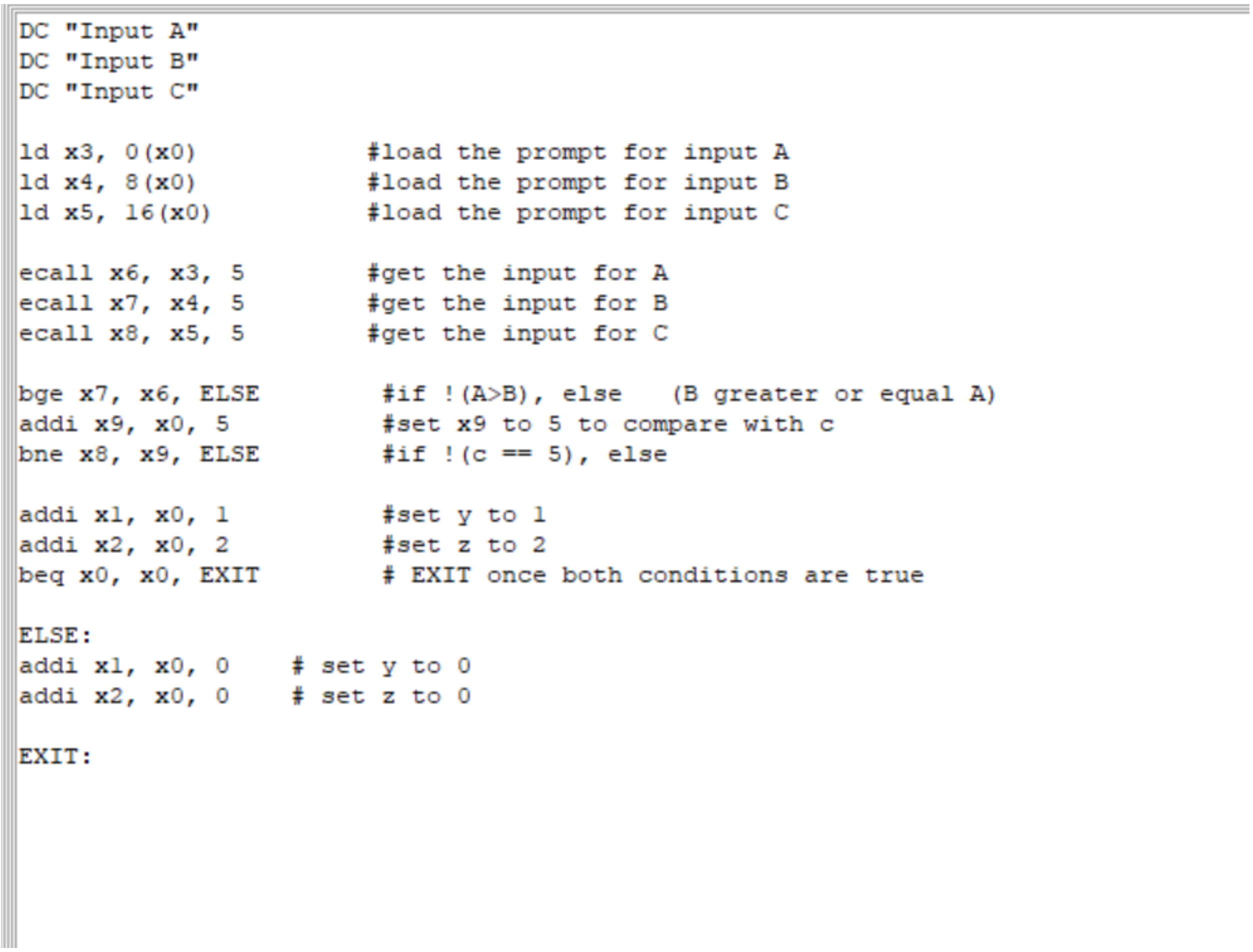
Clear Execution

```
int?3
0x0000000000000003
```

Clear □ OUT

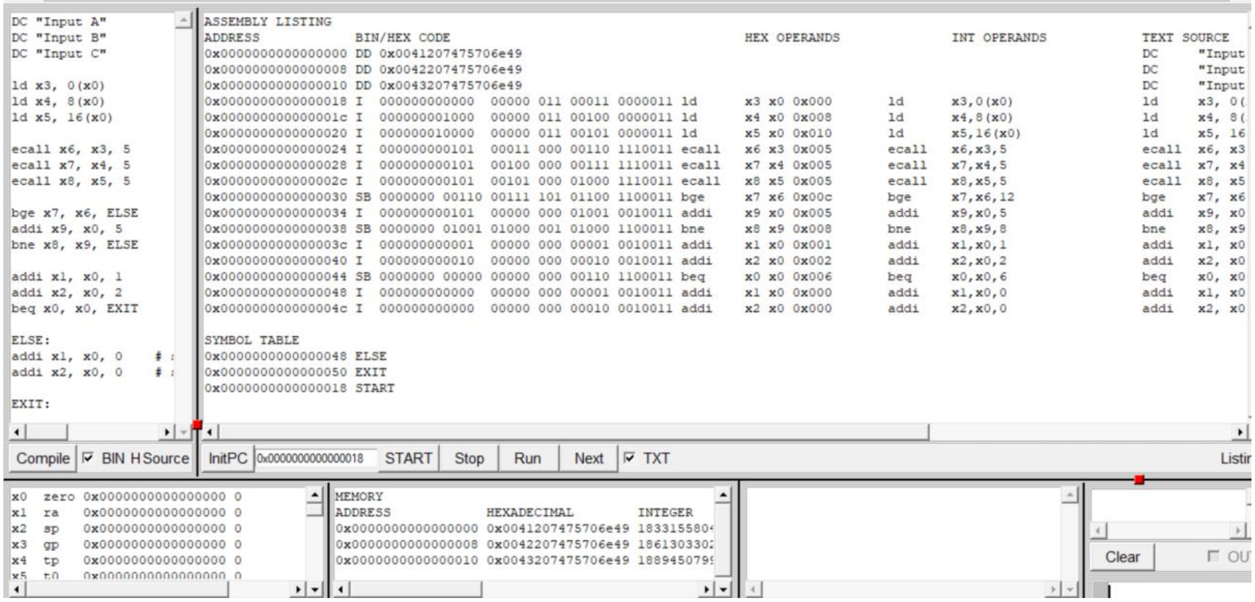
Cancel Enter □ INP

Program 3:



```
DC "Input A"
DC "Input B"
DC "Input C"

ld x3, 0(x0)            #load the prompt for input A
ld x4, 8(x0)            #load the prompt for input B
ld x5, 16(x0)           #load the prompt for input C

ecall x6, x3, 5         #get the input for A
ecall x7, x4, 5         #get the input for B
ecall x8, x5, 5         #get the input for C

bge x7, x6, ELSE         #if !(A>B), else   (B greater or equal A)
addi x9, x0, 5           #set x9 to 5 to compare with c
bne x8, x9, ELSE         #if !(c == 5), else

addi x1, x0, 1           #set y to 1
addi x2, x0, 2           #set z to 2
beq x0, x0, EXIT         # EXIT once both conditions are true

ELSE:
addi x1, x0, 0     # set y to 0
addi x2, x0, 0     # set z to 0

EXIT:
```

After compiling:



```
DC "Input A"
DC "Input B"
DC "Input C"

ld x3, 0(x0)
ld x4, 8(x0)
ld x5, 16(x0)

ecall x6, x3, 5
ecall x7, x4, 5
ecall x8, x5, 5

bge x7, x6, ELSE
addi x9, x0, 5
bne x8, x9, ELSE

addi x1, x0, 1
addi x2, x0, 2
beq x0, x0, EXIT

ELSE:
addi x1, x0, 0      #
addi x2, x0, 0      #

EXIT:
```

```
ASSEMBLY LISTING
ADDRESS                BIN/HEX CODE                                         HEX OPERANDS           INT OPERANDS            TEXT SOURCE
0x0000000000000000 DD 0x0041207475706e49                                                                                  DC     "Input
0x0000000000000008 DD 0x0042207475706e49                                                                                  DC     "Input
0x0000000000000010 DD 0x0043207475706e49                                                                                  DC     "Input
0x0000000000000018 I  000000000000  00000 011 00011 0000011 ld    x3 x0 0x000    ld     x3,0(x0)         ld     x3, 0(
0x000000000000001c I  000000001000  00000 011 00100 0000011 ld    x4 x0 0x008    ld     x4,8(x0)         ld     x4, 8(
0x0000000000000020 I  000000010000  00000 011 00101 0000011 ld    x5 x0 0x010    ld     x5,16(x0)        ld     x5, 16
0x0000000000000024 I  000000000101  00011 000 00110 1110011 ecall x6 x3 0x005    ecall  x6,x3,5          ecall  x6, x3
0x0000000000000028 I  000000000101  00100 000 00111 1110011 ecall x7 x4 0x005    ecall  x7,x4,5          ecall  x7, x4
0x000000000000002c I  000000000101  00101 000 01000 1110011 ecall x8 x5 0x005    ecall  x8,x5,5          ecall  x8, x5
0x0000000000000030 SB 0000000 00110 00111 101 01100 1100011 bge   x7 x6 0x00c    bge    x7,x6,12         bge    x7, x6
0x0000000000000034 I  000000000101  00000 000 01001 0010011 addi  x9 x0 0x005    addi   x9,x0,5          addi   x9, x0
0x0000000000000038 SB 0000000 01001 01000 001 01000 1100011 bne   x8 x9 0x008    bne    x8,x9,8          bne    x8, x9
0x000000000000003c I  000000000001  00000 000 00001 0010011 addi  x1 x0 0x001    addi   x1,x0,1          addi   x1, x0
0x0000000000000040 I  000000000010  00000 000 00010 0010011 addi  x2 x0 0x002    addi   x2,x0,2          addi   x2, x0
0x0000000000000044 SB 0000000 00000 00000 000 00110 1100011 beq   x0 x0 0x006    beq    x0,x0,6          beq    x0, x0
0x0000000000000048 I  000000000000  00000 000 00001 0010011 addi  x1 x0 0x000    addi   x1,x0,0          addi   x1, x0
0x000000000000004c I  000000000000  00000 000 00010 0010011 addi  x2 x0 0x000    addi   x2,x0,0          addi   x2, x0

SYMBOL TABLE
0x0000000000000048 ELSE
0x0000000000000050 EXIT
0x0000000000000018 START
```

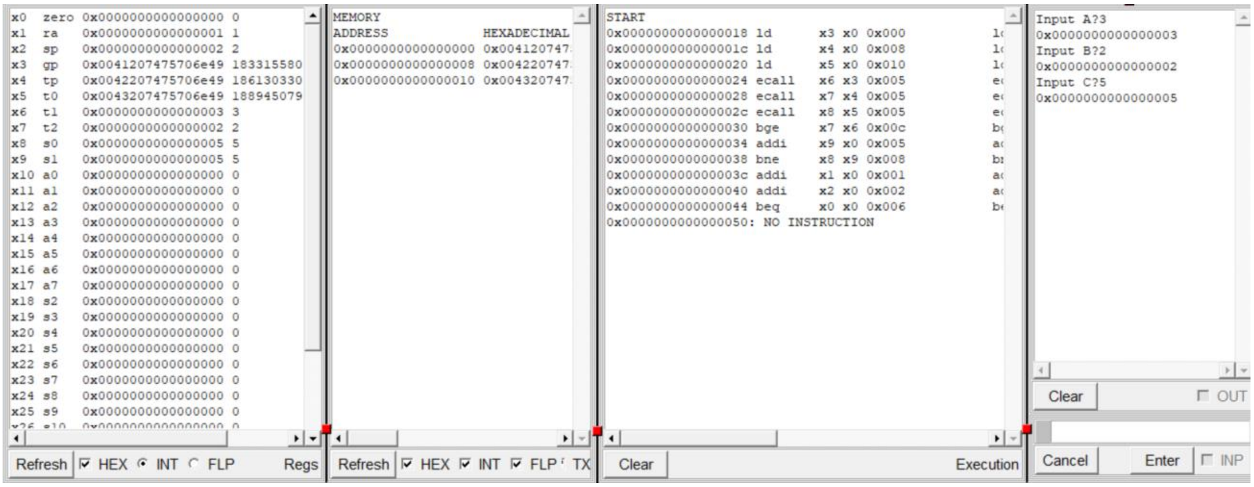
InitPC 0x0000000000000018

```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000000 0
x2  sp   0x0000000000000000 0
x3  gp   0x0000000000000000 0
x4  tp   0x0000000000000000 0
x5  t0   0x0000000000000000 0
```

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER
0x0000000000000000 0x0041207475706e49 183315580
0x0000000000000008 0x0042207475706e49 186130330
0x0000000000000010 0x0043207475706e49 188945079
```

After running:

A>B && C == 5:



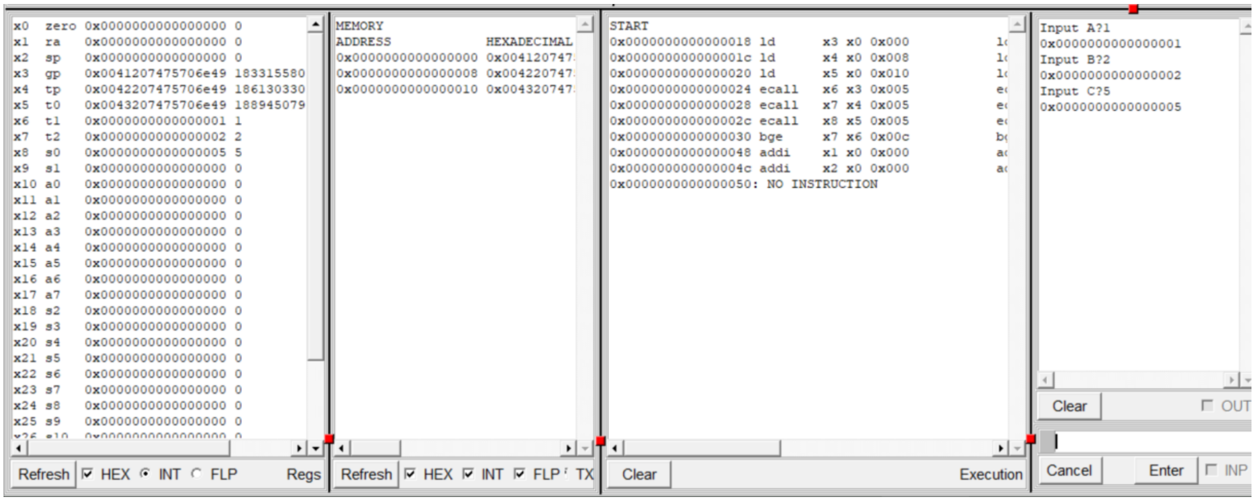
```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000001 1
x2  sp   0x0000000000000002 2
x3  gp   0x0041207475706e49 183315580
x4  tp   0x0042207475706e49 186130330
x5  t0   0x0043207475706e49 188945079
x6  t1   0x0000000000000003 3
x7  t2   0x0000000000000002 2
x8  s0   0x0000000000000005 5
x9  s1   0x0000000000000005 5
x10 a0   0x0000000000000000 0
x11 a1   0x0000000000000000 0
x12 a2   0x0000000000000000 0
x13 a3   0x0000000000000000 0
x14 a4   0x0000000000000000 0
x15 a5   0x0000000000000000 0
x16 a6   0x0000000000000000 0
x17 a7   0x0000000000000000 0
x18 s2   0x0000000000000000 0
x19 s3   0x0000000000000000 0
x20 s4   0x0000000000000000 0
x21 s5   0x0000000000000000 0
x22 s6   0x0000000000000000 0
x23 s7   0x0000000000000000 0
x24 s8   0x0000000000000000 0
x25 s9   0x0000000000000000 0
x26 s10  0x0000000000000000 0
```

```
MEMORY
ADDRESS            HEXADECIMAL
0x0000000000000000 0x004120747
0x0000000000000008 0x004220747
0x0000000000000010 0x004320747
```

```
START
0x0000000000000018 ld      x3 x0 0x000        l
0x000000000000001c ld      x4 x0 0x008        l
0x0000000000000020 ld      x5 x0 0x010        l
0x0000000000000024 ecall   x6 x3 0x005        e
0x0000000000000028 ecall   x7 x4 0x005        e
0x000000000000002c ecall   x8 x5 0x005        e
0x0000000000000030 bge     x7 x6 0x00c        b
0x0000000000000034 addi    x9 x0 0x005        a
0x0000000000000038 bne     x8 x9 0x008        b
0x000000000000003c addi    x1 x0 0x001        a
0x0000000000000040 addi    x2 x0 0x002        a
0x0000000000000044 beq     x0 x0 0x006        b
0x0000000000000050: NO INSTRUCTION
```

```
Input A?3
0x0000000000000003
Input B?2
0x0000000000000002
Input C?5
0x0000000000000005
```

A < B && C == 5



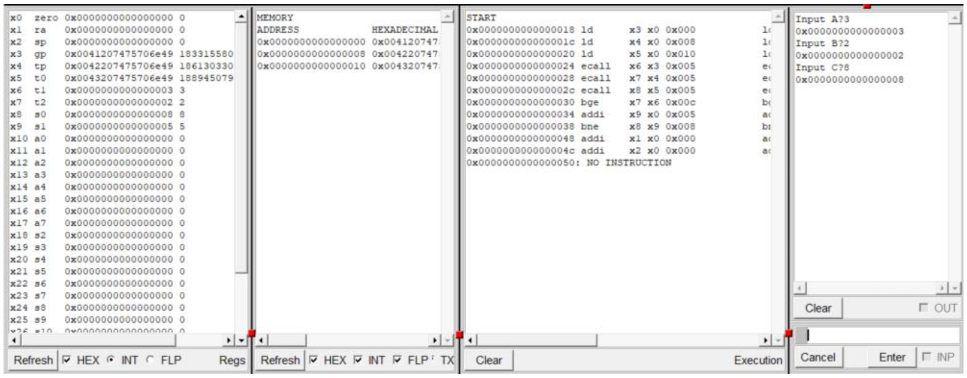
```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000000 0
x2  sp   0x0000000000000000 0
x3  gp   0x0041207475706e49 183315580
x4  tp   0x0042207475706e49 186130330
x5  t0   0x0043207475706e49 188945079
x6  t1   0x0000000000000001 1
x7  t2   0x0000000000000002 2
x8  s0   0x0000000000000005 5
x9  s1   0x0000000000000000 0
x10 a0   0x0000000000000000 0
x11 a1   0x0000000000000000 0
x12 a2   0x0000000000000000 0
x13 a3   0x0000000000000000 0
x14 a4   0x0000000000000000 0
x15 a5   0x0000000000000000 0
x16 a6   0x0000000000000000 0
x17 a7   0x0000000000000000 0
x18 s2   0x0000000000000000 0
x19 s3   0x0000000000000000 0
x20 s4   0x0000000000000000 0
x21 s5   0x0000000000000000 0
x22 s6   0x0000000000000000 0
x23 s7   0x0000000000000000 0
x24 s8   0x0000000000000000 0
x25 s9   0x0000000000000000 0
x26 s10  0x0000000000000000 0
```

```
MEMORY
ADDRESS             HEXADECIMAL
0x0000000000000000  0x004120747
0x0000000000000008  0x004220747
0x0000000000000010  0x004320747
```

```
START
0x0000000000000018 ld       x3 x0 0x000      l
0x000000000000001c ld       x4 x0 0x008      l
0x0000000000000020 ld       x5 x0 0x010      l
0x0000000000000024 ecall    x6 x3 0x005      e
0x0000000000000028 ecall    x7 x4 0x005      e
0x000000000000002c ecall    x8 x5 0x005      e
0x0000000000000030 bge      x7 x6 0x00c      b
0x0000000000000048 addi     x1 x0 0x000      a
0x000000000000004c addi     x2 x0 0x000      a
0x0000000000000050: NO INSTRUCTION
```

```
Input A?1
0x0000000000000001
Input B?2
0x0000000000000002
Input C?5
0x0000000000000005
```

A>B && C != 5



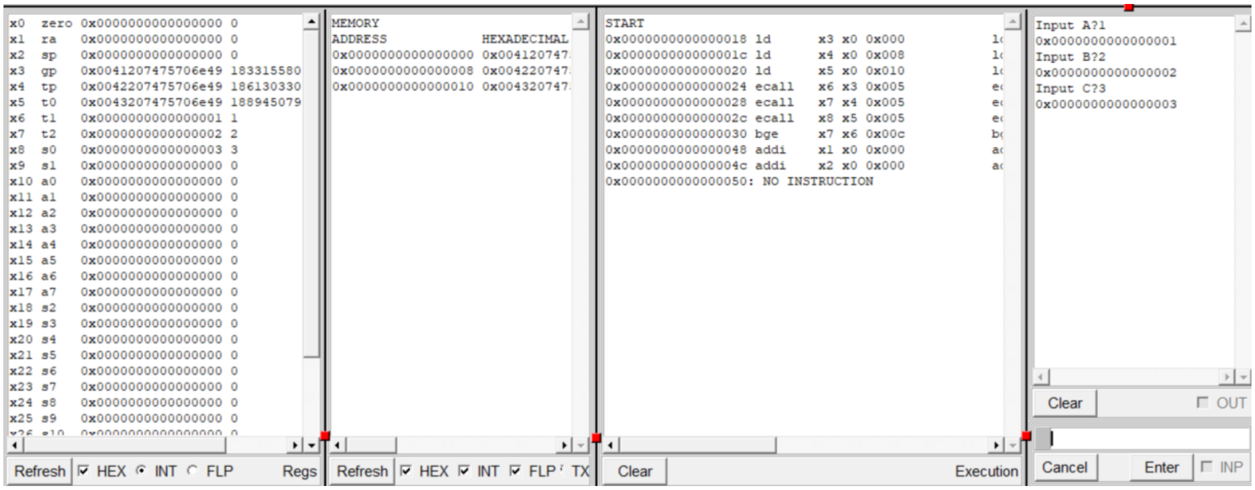
```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000000 0
x2  sp   0x0000000000000000 0
x3  gp   0x0041207475706e49 183315580
x4  tp   0x0042207475706e49 186130330
x5  t0   0x0043207475706e49 188945079
x6  t1   0x0000000000000003 3
x7  t2   0x0000000000000002 2
x8  s0   0x0000000000000008 8
x9  s1   0x0000000000000005 5
x10 a0   0x0000000000000000 0
x11 a1   0x0000000000000000 0
x12 a2   0x0000000000000000 0
x13 a3   0x0000000000000000 0
x14 a4   0x0000000000000000 0
x15 a5   0x0000000000000000 0
x16 a6   0x0000000000000000 0
x17 a7   0x0000000000000000 0
x18 s2   0x0000000000000000 0
x19 s3   0x0000000000000000 0
x20 s4   0x0000000000000000 0
x21 s5   0x0000000000000000 0
x22 s6   0x0000000000000000 0
x23 s7   0x0000000000000000 0
x24 s8   0x0000000000000000 0
x25 s9   0x0000000000000000 0
x26 s10  0x0000000000000000 0
```

```
MEMORY
ADDRESS             HEXADECIMAL
0x0000000000000000  0x004120747
0x0000000000000008  0x004220747
0x0000000000000010  0x004320747
```

```
START
0x0000000000000018 ld       x3 x0 0x000      l
0x000000000000001c ld       x4 x0 0x008      l
0x0000000000000020 ld       x5 x0 0x010      l
0x0000000000000024 ecall    x6 x3 0x005      e
0x0000000000000028 ecall    x7 x4 0x005      e
0x000000000000002c ecall    x8 x5 0x005      e
0x0000000000000030 bge      x7 x6 0x00c      b
0x0000000000000034 addi     x9 x0 0x005      a
0x0000000000000038 bne      x8 x9 0x008      b
0x0000000000000048 addi     x1 x0 0x000      a
0x000000000000004c addi     x2 x0 0x000      a
0x0000000000000050: NO INSTRUCTION
```

```
Input A?3
0x0000000000000003
Input B?2
0x0000000000000002
Input C?8
0x0000000000000008
```

A < B && C != 5:



```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000000 0
x2  sp   0x0000000000000000 0
x3  gp   0x0041207475706e49 183315580
x4  tp   0x0042207475706e49 186130330
x5  t0   0x0043207475706e49 188945079
x6  t1   0x0000000000000001 1
x7  t2   0x0000000000000002 2
x8  s0   0x0000000000000003 3
x9  s1   0x0000000000000000 0
x10 a0   0x0000000000000000 0
x11 a1   0x0000000000000000 0
x12 a2   0x0000000000000000 0
x13 a3   0x0000000000000000 0
x14 a4   0x0000000000000000 0
x15 a5   0x0000000000000000 0
x16 a6   0x0000000000000000 0
x17 a7   0x0000000000000000 0
x18 s2   0x0000000000000000 0
x19 s3   0x0000000000000000 0
x20 s4   0x0000000000000000 0
x21 s5   0x0000000000000000 0
x22 s6   0x0000000000000000 0
x23 s7   0x0000000000000000 0
x24 s8   0x0000000000000000 0
x25 s9   0x0000000000000000 0
x26 s10  0x0000000000000000 0
```

```
MEMORY
ADDRESS             HEXADECIMAL
0x0000000000000000  0x004120747
0x0000000000000008  0x004220747
0x0000000000000010  0x004320747
```

```
START
0x0000000000000018 ld       x3 x0 0x000      l
0x000000000000001c ld       x4 x0 0x008      l
0x0000000000000020 ld       x5 x0 0x010      l
0x0000000000000024 ecall    x6 x3 0x005      e
0x0000000000000028 ecall    x7 x4 0x005      e
0x000000000000002c ecall    x8 x5 0x005      e
0x0000000000000030 bge      x7 x6 0x00c      b
0x0000000000000048 addi     x1 x0 0x000      a
0x000000000000004c addi     x2 x0 0x000      a
0x0000000000000050: NO INSTRUCTION
```

```
Input A?1
0x0000000000000001
Input B?2
0x0000000000000002
Input C?3
0x0000000000000003
```