Bottom text

Lab B

Sample text

Jonathan Lai
214838445

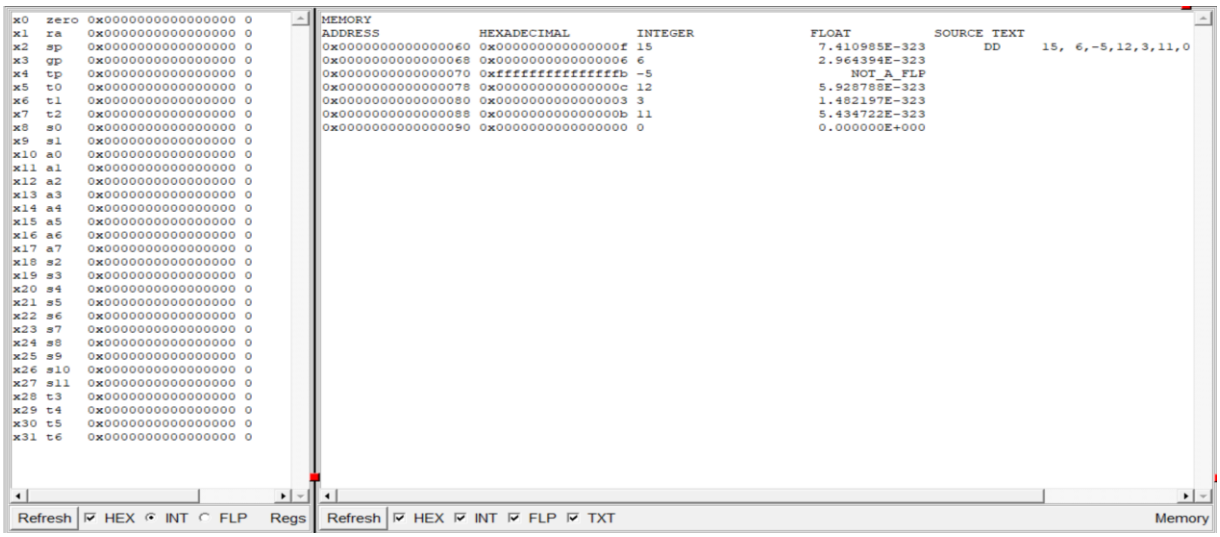
Program 1:

Code:

```
ORG 96
DD 15, 6,-5,12,3,11,0
addi x1, x0, 96

ld x4, 0(x1) //y
ld x5, 8(x1) //z
ld x6, 16(x1) //c
ld x7, 24(x1)//d
ld x8, 32(x1)//L
ld x9, 40(x1) //M
ld x10, 48(x1)//x


add x11, x4, x9 //y+m

sub x12, x8, x7 // L-D

add x13, x5, x6 //z+c

add x10, x0, x11
sub x10, x10, x12
add x10, x10, x13
sub x10, x10, x7

sd x10, 48(x1)
```

After compile (Before run):

Register/Memory



Listing

```
ASSEMBLY LISTING
ADDRESS           BIN/HEX CODE                                HEX OPERANDS            INT OPERANDS            TEXT SOURCE
0x0000000000000060 DD 0x000000000000000f                                                                      DD     15, 6,-5,12,3,11,0
0x0000000000000068 DD 0x0000000000000006
0x0000000000000070 DD 0xfffffffffffffffb
0x0000000000000078 DD 0x000000000000000c
0x0000000000000080 DD 0x0000000000000003
0x0000000000000088 DD 0x000000000000000b
0x0000000000000090 DD 0x0000000000000000
0x0000000000000098 I  000001100000  00000 000 00001 0010011 addi    x1 x0 0x060       addi    x1,x0,96              addi    x1, x0, 96
0x000000000000009c I  000000000000  00001 011 00100 0000011 ld      x4 x1 0x000       ld      x4,0(x1)              ld      x4, 0(x1)           //y
0x00000000000000a0 I  000000001000  00001 011 00101 0000011 ld      x5 x1 0x008       ld      x5,8(x1)              ld      x5, 8(x1)           //z
0x00000000000000a4 I  000000010000  00001 011 00110 0000011 ld      x6 x1 0x010       ld      x6,16(x1)             ld      x6, 16(x1)          //c
0x00000000000000a8 I  000000011000  00001 011 00111 0000011 ld      x7 x1 0x018       ld      x7,24(x1)             ld      x7, 24(x1)          //d
0x00000000000000ac I  000000100000  00001 011 01000 0000011 ld      x8 x1 0x020       ld      x8,32(x1)             ld      x8, 32(x1)          //L
0x00000000000000b0 I  000000101000  00001 011 01001 0000011 ld      x9 x1 0x028       ld      x9,40(x1)             ld      x9, 40(x1)          //M
0x00000000000000b4 I  000000110000  00001 011 01010 0000011 ld      x10 x1 0x030      ld      x10,48(x1)            ld      x10, 48(x1)         //x
0x00000000000000b8 R  0000000 01001 00100 000 01011 0110011 add     x11 x4 x9         add     x11,x4,x9             add     x11, x4, x9         //y+m
0x00000000000000bc R  0100000 00111 01000 000 01100 0110011 sub     x12 x8 x7         sub     x12,x8,x7             sub     x12, x8, x7         // L-D
0x00000000000000c0 R  0000000 00110 00101 000 01101 0110011 add     x13 x5 x6         add     x13,x5,x6             add     x13, x5, x6         //z+c
0x00000000000000c4 R  0000000 01011 00000 000 01010 0110011 add     x10 x0 x11        add     x10,x0,x11            add     x10, x0, x11
0x00000000000000c8 R  0100000 01100 01010 000 01010 0110011 sub     x10 x10 x12       sub     x10,x10,x12           sub     x10, x10, x12
0x00000000000000cc R  0000000 01101 01010 000 01010 0110011 add     x10 x10 x13       add     x10,x10,x13           add     x10, x10, x13
0x00000000000000d0 R  0100000 00111 01010 000 01010 0110011 sub     x10 x10 x7        sub     x10,x10,x7            sub     x10, x10, x7
0x00000000000000d4 S  0000001 01010 00001 011 10000 0100011 sd      x10 x1 0x030      sd      x10,48(x1)            sd      x10, 48(x1)

SYMBOL TABLE
0x0000000000000098 START
```

InitPC | 0x0000000000000098 | START | Stop | Run | Next | ☑ TXT | Listing

After Run:

Register/Memory

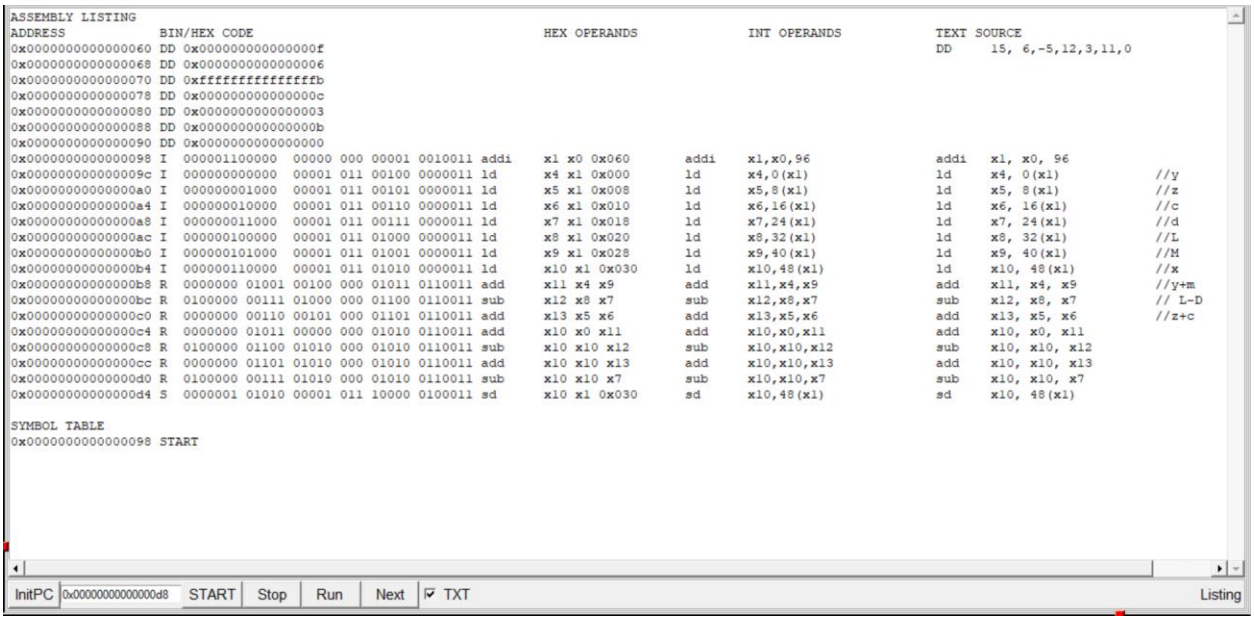
```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000060 96
x2  sp   0x0000000000000000 0
x3  gp   0x0000000000000000 0
x4  tp   0x000000000000000f 15
x5  t0   0x0000000000000006 6
x6  t1   0xfffffffffffffffb -5
x7  t2   0x000000000000000c 12
x8  s0   0x0000000000000003 3
x9  s1   0x000000000000000b 11
x10 a0   0x0000000000000018 24
x11 a1   0x000000000000001A 26
x12 a2   0xFFFFFFFFFFFFFF7 -9
x13 a3   0x0000000000000001 1
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
x27 s11  0x0000000000000000 0
x28 t3   0x0000000000000000 0
x29 t4   0x0000000000000000 0
x30 t5   0x0000000000000000 0
x31 t6   0x0000000000000000 0
```

Refresh | ☑ HEX | ◉ INT | ○ FLP | Regs

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER              FLOAT          SOURCE TEXT
0x0000000000000060 0x000000000000000f 15                    7.410985E-323       DD     15, 6,-5,12,3,11,0
0x0000000000000068 0x0000000000000006 6                     2.964394E-323
0x0000000000000070 0xfffffffffffffffb -5                        NOT_A_FLP
0x0000000000000078 0x000000000000000c 12                    5.928788E-323
0x0000000000000080 0x0000000000000003 3                     1.482197E-323
0x0000000000000088 0x000000000000000b 11                    5.434722E-323
0x0000000000000090 0x0000000000000018 24                    1.185758E-322
```

Refresh | ☑ HEX | ☑ INT | ☑ FLP | ☑ TXT | Memory

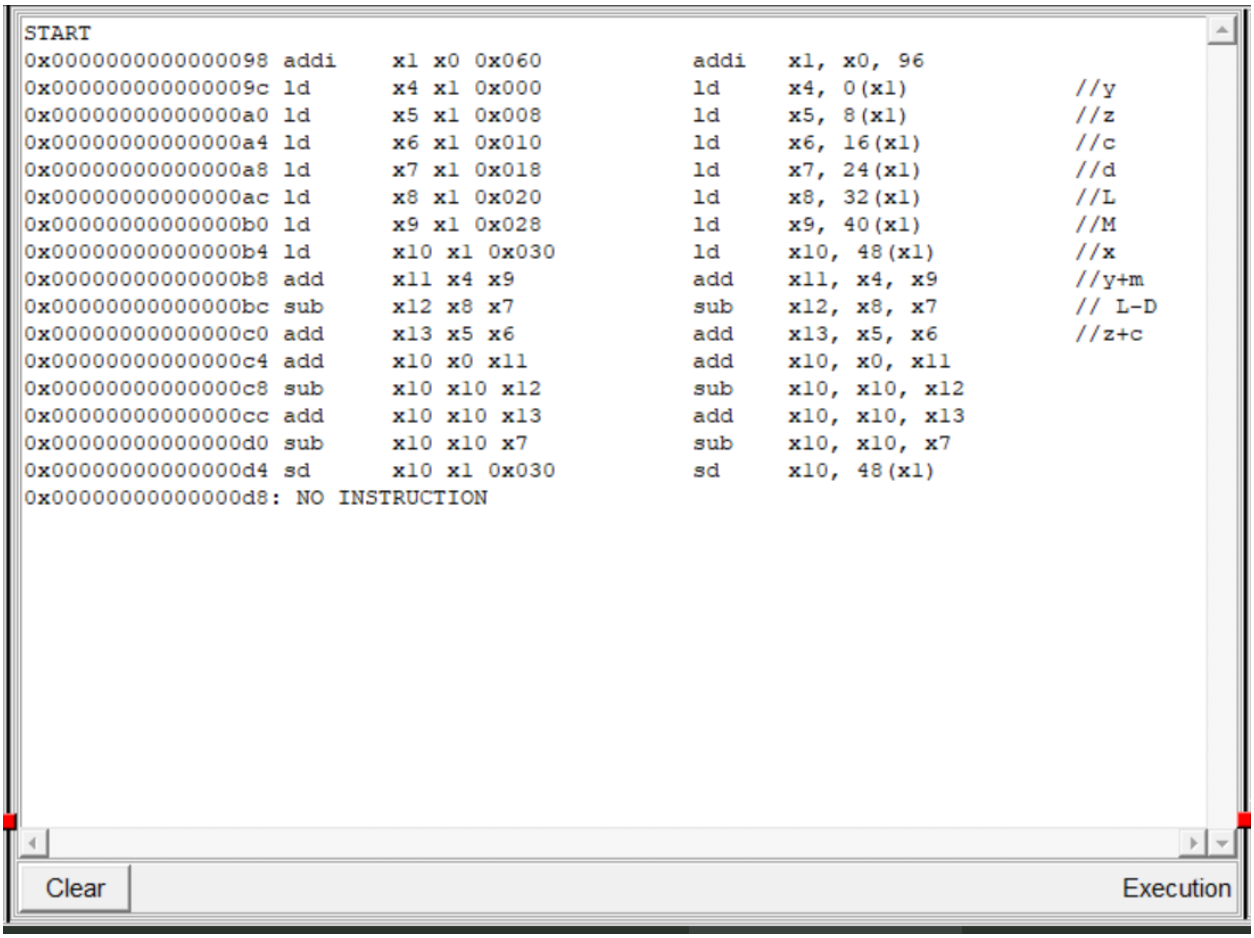
Listing



```
ASSEMBLY LISTING
ADDRESS          BIN/HEX CODE                                HEX OPERANDS            INT OPERANDS            TEXT SOURCE
0x0000000000000060 DD 0x000000000000000f                                                                     DD     15, 6,-5,12,3,11,0
0x0000000000000068 DD 0x0000000000000006
0x0000000000000070 DD 0xfffffffffffffffb
0x0000000000000078 DD 0x000000000000000c
0x0000000000000080 DD 0x0000000000000003
0x0000000000000088 DD 0x000000000000000b
0x0000000000000090 DD 0x0000000000000000
0x0000000000000098 I  000001100000  00000 000 00001 0010011 addi    x1 x0 0x060       addi    x1,x0,96            addi   x1, x0, 96
0x000000000000009c I  000000000000  00001 011 00100 0000011 ld      x4 x1 0x000       ld      x4,0(x1)            ld     x4, 0(x1)          //y
0x00000000000000a0 I  000000001000  00001 011 00101 0000011 ld      x5 x1 0x008       ld      x5,8(x1)            ld     x5, 8(x1)          //z
0x00000000000000a4 I  000000010000  00001 011 00110 0000011 ld      x6 x1 0x010       ld      x6,16(x1)           ld     x6, 16(x1)         //c
0x00000000000000a8 I  000000011000  00001 011 00111 0000011 ld      x7 x1 0x018       ld      x7,24(x1)           ld     x7, 24(x1)         //d
0x00000000000000ac I  000000100000  00001 011 01000 0000011 ld      x8 x1 0x020       ld      x8,32(x1)           ld     x8, 32(x1)         //L
0x00000000000000b0 I  000000101000  00001 011 01001 0000011 ld      x9 x1 0x028       ld      x9,40(x1)           ld     x9, 40(x1)         //M
0x00000000000000b4 I  000000110000  00001 011 01010 0000011 ld      x10 x1 0x030      ld      x10,48(x1)          ld     x10, 48(x1)        //x
0x00000000000000b8 R  0000000 01001 00100 000 01011 0110011 add     x11 x4 x9         add     x11,x4,x9           add    x11, x4, x9        //y+m
0x00000000000000bc R  0100000 00111 01000 000 01100 0110011 sub     x12 x8 x7         sub     x12,x8,x7           sub    x12, x8, x7        // L-D
0x00000000000000c0 R  0000000 00110 00101 000 01101 0110011 add     x13 x5 x6         add     x13,x5,x6           add    x13, x5, x6        //z+c
0x00000000000000c4 R  0000000 01011 00000 000 01010 0110011 add     x10 x0 x11        add     x10,x0,x11          add    x10, x0, x11
0x00000000000000c8 R  0100000 01100 01010 000 01010 0110011 sub     x10 x10 x12       sub     x10,x10,x12         sub    x10, x10, x12
0x00000000000000cc R  0000000 01101 01010 000 01010 0110011 add     x10 x10 x13       add     x10,x10,x13         add    x10, x10, x13
0x00000000000000d0 R  0100000 00111 01010 000 01010 0110011 sub     x10 x10 x7        sub     x10,x10,x7          sub    x10, x10, x7
0x00000000000000d4 S  0000001 01010 00001 011 10000 0100011 sd      x10 x1 0x030      sd      x10,48(x1)          sd     x10, 48(x1)

SYMBOL TABLE
0x0000000000000098 START
```

InitPC | 0x00000000000000d8 | START | Stop | Run | Next | TXT | Listing

Execution



```
START
0x0000000000000098 addi    x1 x0 0x060          addi   x1, x0, 96
0x000000000000009c ld      x4 x1 0x000          ld     x4, 0(x1)          //y
0x00000000000000a0 ld      x5 x1 0x008          ld     x5, 8(x1)          //z
0x00000000000000a4 ld      x6 x1 0x010          ld     x6, 16(x1)         //c
0x00000000000000a8 ld      x7 x1 0x018          ld     x7, 24(x1)         //d
0x00000000000000ac ld      x8 x1 0x020          ld     x8, 32(x1)         //L
0x00000000000000b0 ld      x9 x1 0x028          ld     x9, 40(x1)         //M
0x00000000000000b4 ld      x10 x1 0x030         ld     x10, 48(x1)        //x
0x00000000000000b8 add     x11 x4 x9            add    x11, x4, x9        //y+m
0x00000000000000bc sub     x12 x8 x7            sub    x12, x8, x7        // L-D
0x00000000000000c0 add     x13 x5 x6            add    x13, x5, x6        //z+c
0x00000000000000c4 add     x10 x0 x11           add    x10, x0, x11
0x00000000000000c8 sub     x10 x10 x12          sub    x10, x10, x12
0x00000000000000cc add     x10 x10 x13          add    x10, x10, x13
0x00000000000000d0 sub     x10 x10 x7           sub    x10, x10, x7
0x00000000000000d4 sd      x10 x1 0x030         sd     x10, 48(x1)
0x00000000000000d8: NO INSTRUCTION
```

Clear | Execution

Program v2:

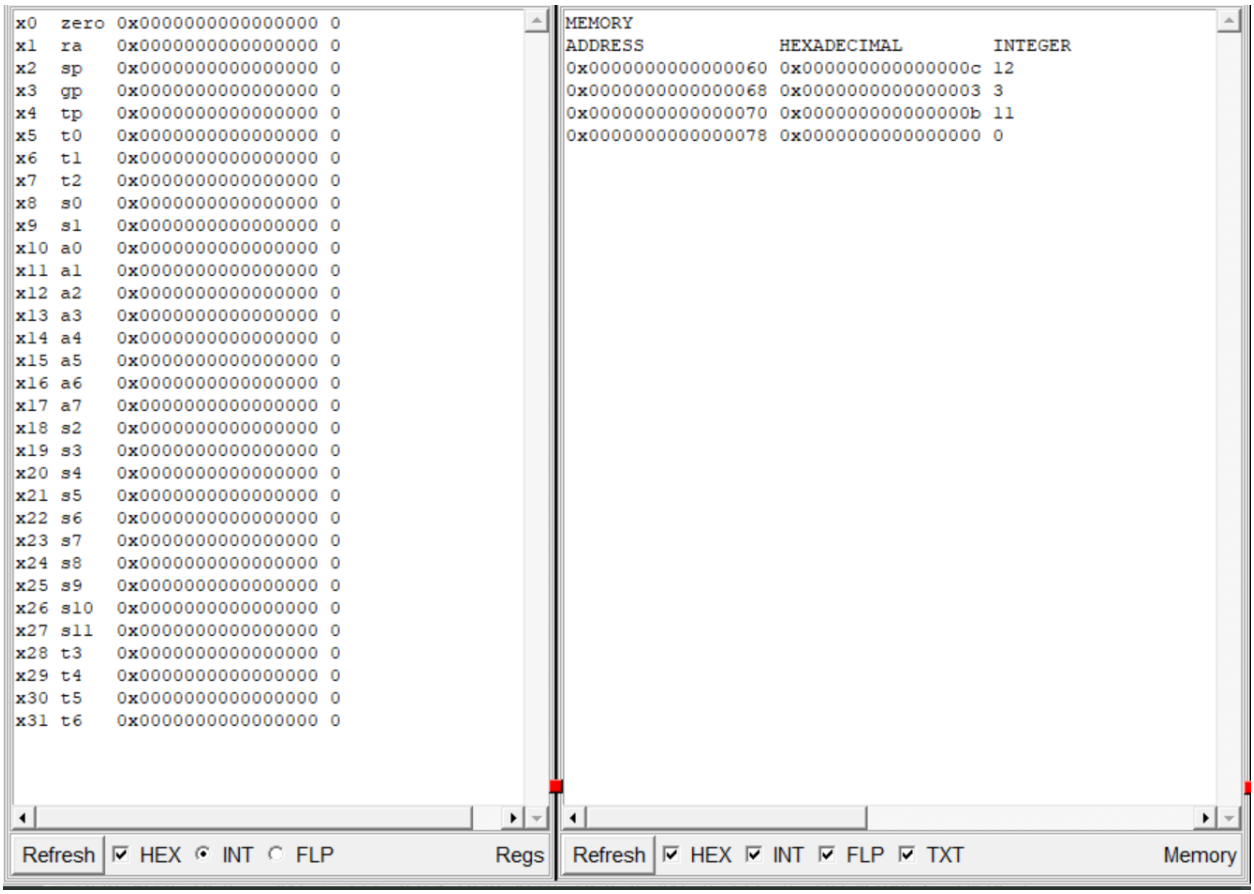
Code:

```
ORG 96
DD 12,3,11,0
addi x1, x0, 96

addi x2, x0, 15 //y
addi x3, x0, 6 //z
addi x4, x0, -5 //c


ld x5, 0(x1) //D
ld x6, 8(x1) //L
ld x7, 16(x1) //M

addi x8, x0, 0 //X

add x9, x2, x7 //Y + M
sub x10, x6, x5 //L -D
add x11, x3 , x4 // Z + C

add x8, x0, x9
sub x8, x8, x10
add x8, x8, x11
sub x8, x8, x5

sd x8, 24(x1)
```

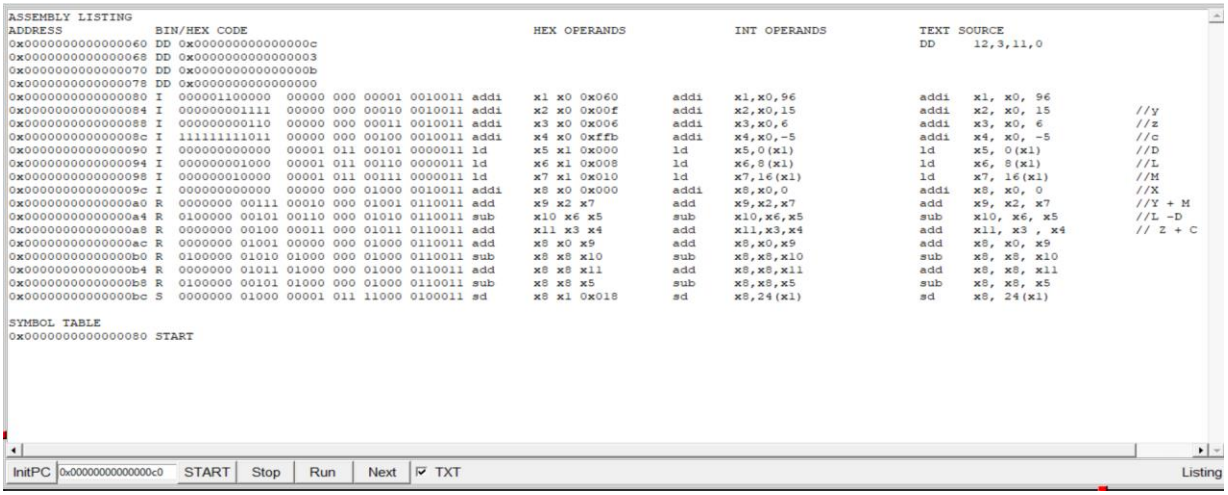
Compile ☑ BIN ☑ HEX ☑ INT ☑ TXT Source

Before Run(After compile):

Register/Memory

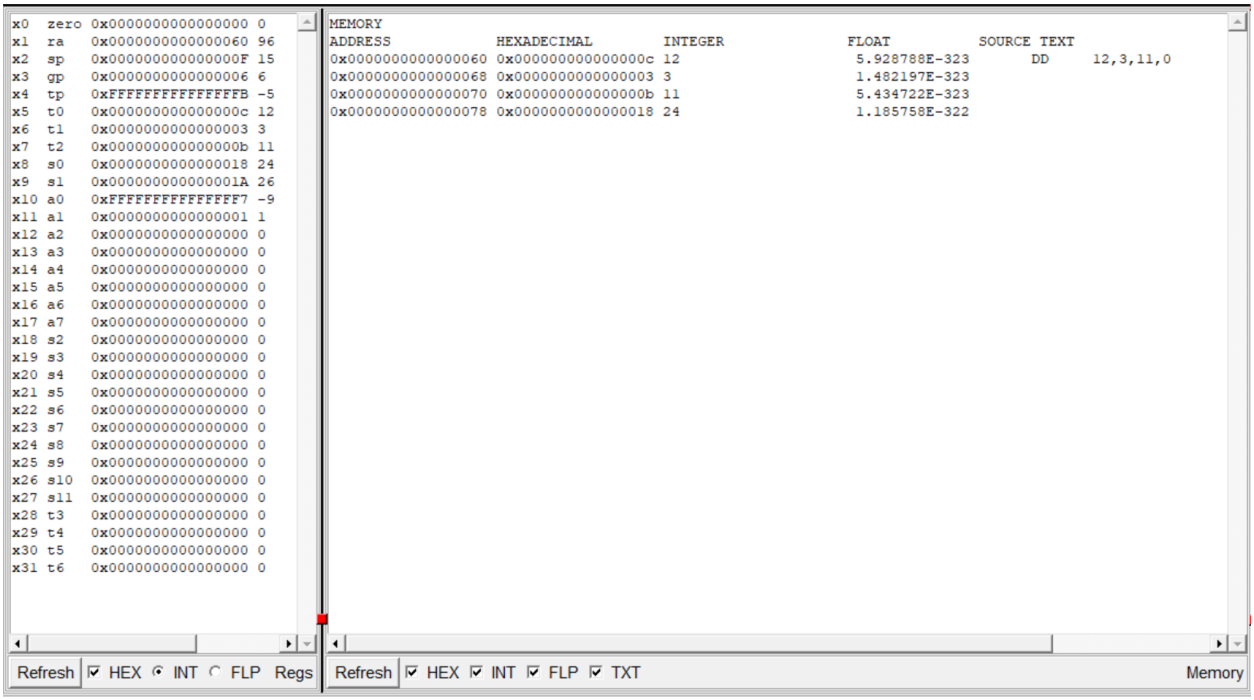


Listing Window:



After run:

Memory/Register Window:



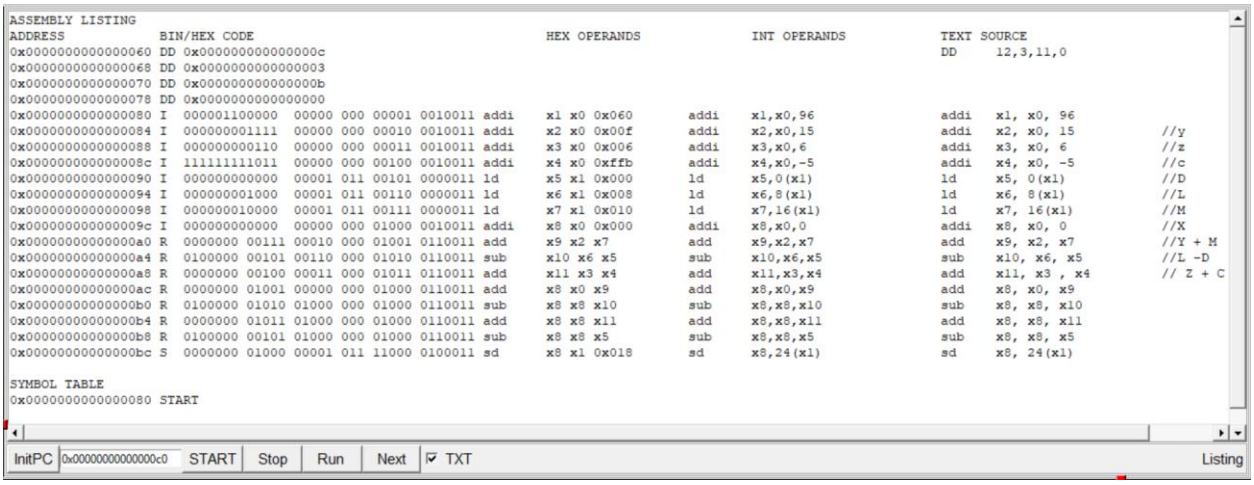
```
x0  zero 0x0000000000000000 0
x1  ra   0x0000000000000060 96
x2  sp   0x000000000000000F 15
x3  gp   0x0000000000000006 6
x4  tp   0xFFFFFFFFFFFFFFFB -5
x5  t0   0x000000000000000c 12
x6  t1   0x0000000000000003 3
x7  t2   0x000000000000000b 11
x8  s0   0x0000000000000018 24
x9  s1   0x000000000000001A 26
x10 a0   0xFFFFFFFFFFFFFFF7 -9
x11 a1   0x0000000000000001 1
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
x27 s11  0x0000000000000000 0
x28 t3   0x0000000000000000 0
x29 t4   0x0000000000000000 0
x30 t5   0x0000000000000000 0
x31 t6   0x0000000000000000 0
```

Refresh ☑ HEX ◉ INT ○ FLP Regs

```
MEMORY
ADDRESS            HEXADECIMAL        INTEGER              FLOAT          SOURCE TEXT
0x0000000000000060 0x000000000000000c 12                    5.928788E-323       DD     12,3,11,0
0x0000000000000068 0x0000000000000003 3                     1.482197E-323
0x0000000000000070 0x000000000000000b 11                    5.434722E-323
0x0000000000000078 0x0000000000000018 24                    1.185758E-322
```

Refresh ☑ HEX ☑ INT ☑ FLP ☑ TXT Memory

Listing



```
ASSEMBLY LISTING
ADDRESS            BIN/HEX CODE                                HEX OPERANDS            INT OPERANDS            TEXT SOURCE
0x0000000000000060 DD 0x000000000000000c                                                                       DD     12,3,11,0
0x0000000000000068 DD 0x0000000000000003
0x0000000000000070 DD 0x000000000000000b
0x0000000000000078 DD 0x0000000000000000
0x0000000000000080 I  000001100000  00000 000 00001 0010011 addi    x1 x0 0x060       addi    x1,x0,96        addi   x1, x0, 96
0x0000000000000084 I  000000001111  00000 000 00010 0010011 addi    x2 x0 0x00f       addi    x2,x0,15        addi   x2, x0, 15          //y
0x0000000000000088 I  000000000110  00000 000 00011 0010011 addi    x3 x0 0x006       addi    x3,x0,6         addi   x3, x0, 6           //z
0x000000000000008c I  111111111011  00000 000 00100 0010011 addi    x4 x0 0xffb       addi    x4,x0,-5        addi   x4, x0, -5          //c
0x0000000000000090 I  000000000000  00001 011 00101 0000011 ld      x5 x1 0x000       ld      x5,0(x1)        ld     x5, 0(x1)           //D
0x0000000000000094 I  000000001000  00001 011 00110 0000011 ld      x6 x1 0x008       ld      x6,8(x1)        ld     x6, 8(x1)           //L
0x0000000000000098 I  000000010000  00001 011 00111 0000011 ld      x7 x1 0x010       ld      x7,16(x1)       ld     x7, 16(x1)          //M
0x000000000000009c I  000000000000  00000 000 01000 0010011 addi    x8 x0 0x000       addi    x8,x0,0         addi   x8, x0, 0           //X
0x00000000000000a0 R  0000000 00111 00010 000 01001 0110011 add     x9 x2 x7          add     x9,x2,x7        add    x9, x2, x7          //Y + M
0x00000000000000a4 R  0100000 00101 00110 000 01010 0110011 sub     x10 x6 x5         sub     x10,x6,x5       sub    x10, x6, x5         //L -D
0x00000000000000a8 R  0000000 00100 00011 000 01011 0110011 add     x11 x3 x4         add     x11,x3,x4       add    x11, x3 , x4        // Z + C
0x00000000000000ac R  0000000 01001 00000 000 01000 0110011 add     x8 x0 x9          add     x8,x0,x9        add    x8, x0, x9
0x00000000000000b0 R  0100000 01010 01000 000 01000 0110011 sub     x8 x8 x10         sub     x8,x8,x10       sub    x8, x8, x10
0x00000000000000b4 R  0000000 01011 01000 000 01000 0110011 add     x8 x8 x11         add     x8,x8,x11       add    x8, x8, x11
0x00000000000000b8 R  0100000 00101 01000 000 01000 0110011 sub     x8 x8 x5          sub     x8,x8,x5        sub    x8, x8, x5
0x00000000000000bc S  0000000 01000 00001 011 11000 0100011 sd      x8 x1 0x018       sd      x8,24(x1)       sd     x8, 24(x1)

SYMBOL TABLE
0x0000000000000080 START
```

InitPC 0x00000000000000c0 START Stop Run Next ☑ TXT Listing

Execution Window