1. Write an assembly program that implements the following algorithm:

x ← 2

if (x != 0)

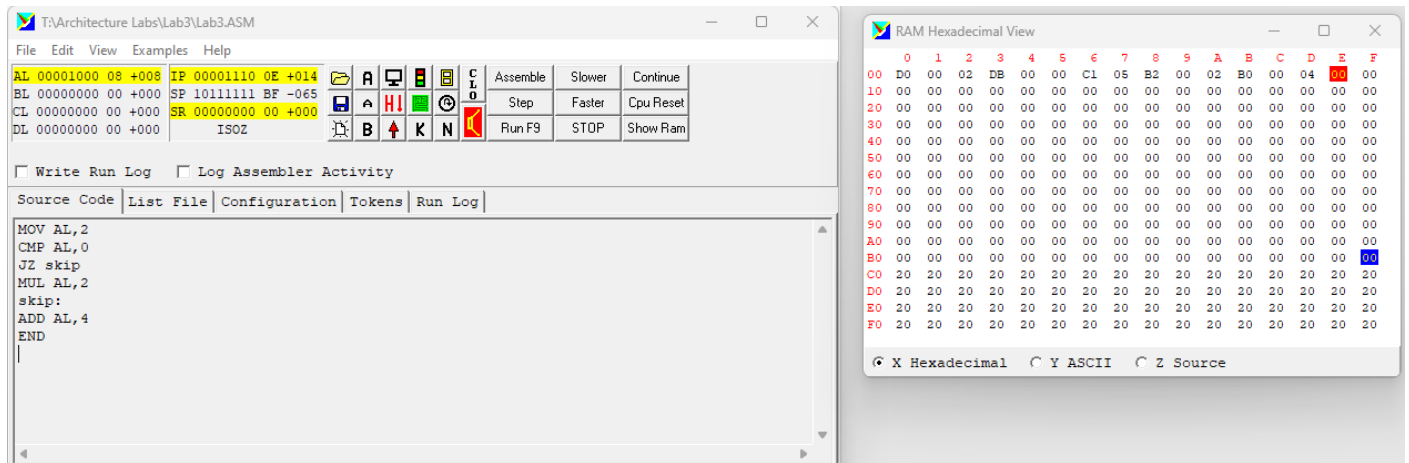{

       x ← x * 2

}

x ← x + 4

The same register that was used to store the input x should also be used to store the output x. Make sure to test the code for different starting values of x to ensure the if statement is working correctly.

2. Write an assembly program that implements the following algorithm:

$x \leftarrow 5$
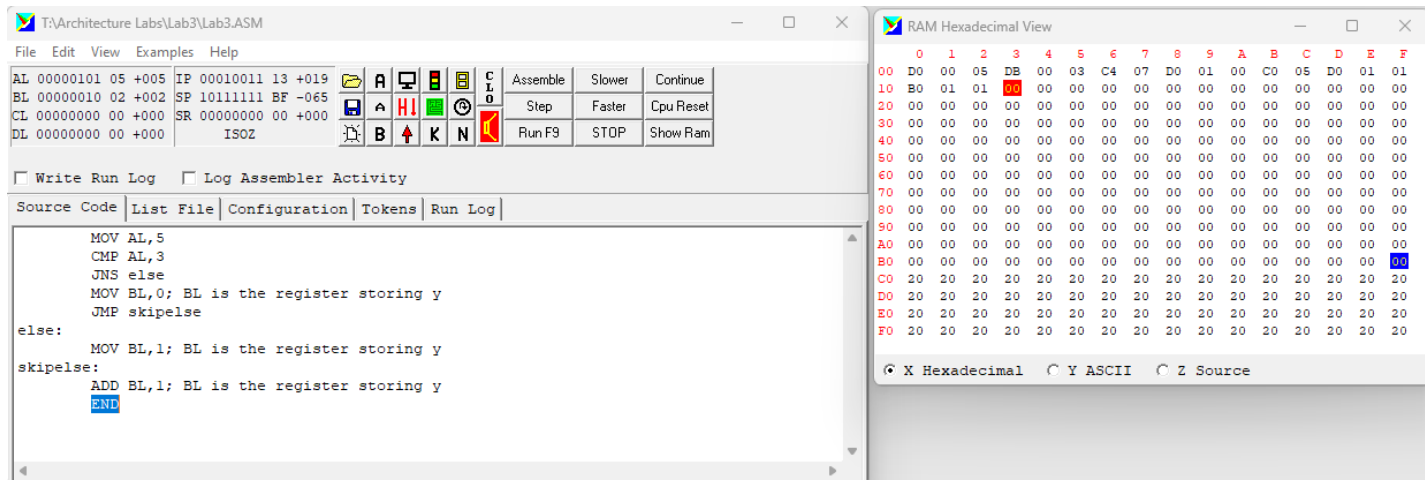
if $(x < 3)$

{

    $y \leftarrow 0$

}

else

{

    $y \leftarrow 1$

}

$y \leftarrow y + 1$

Clearly mark which register you are using to store the output y. Make sure to test the code for different starting values of x to ensure the if-else statement is working correctly.


```
        MOV AL,5
        CMP AL,3
        JNS else
        MOV BL,0; BL is the register storing y
        JMP skipelse
else:
        MOV BL,1; BL is the register storing y
skipelse:
        ADD BL,1; BL is the register storing y
        END
```

3. Write an assembly program that implements the following algorithm:

x ← 4

y ← 1

while (x > 0)

{

       y ← y * 2

       x ← x - 1

}

Clearly mark which register you are using to store the output y. Make sure to test the code for different starting values of x to ensure the while loop is working correctly.

Source Code:

```
MOV AL,4
MOV BL,1; BL is the register storing y
Loop:
        MUL BL,2; BL is the register storing y
        SUB AL,1
        JZ loopend
        JMP Loop
loopend:
        END
```