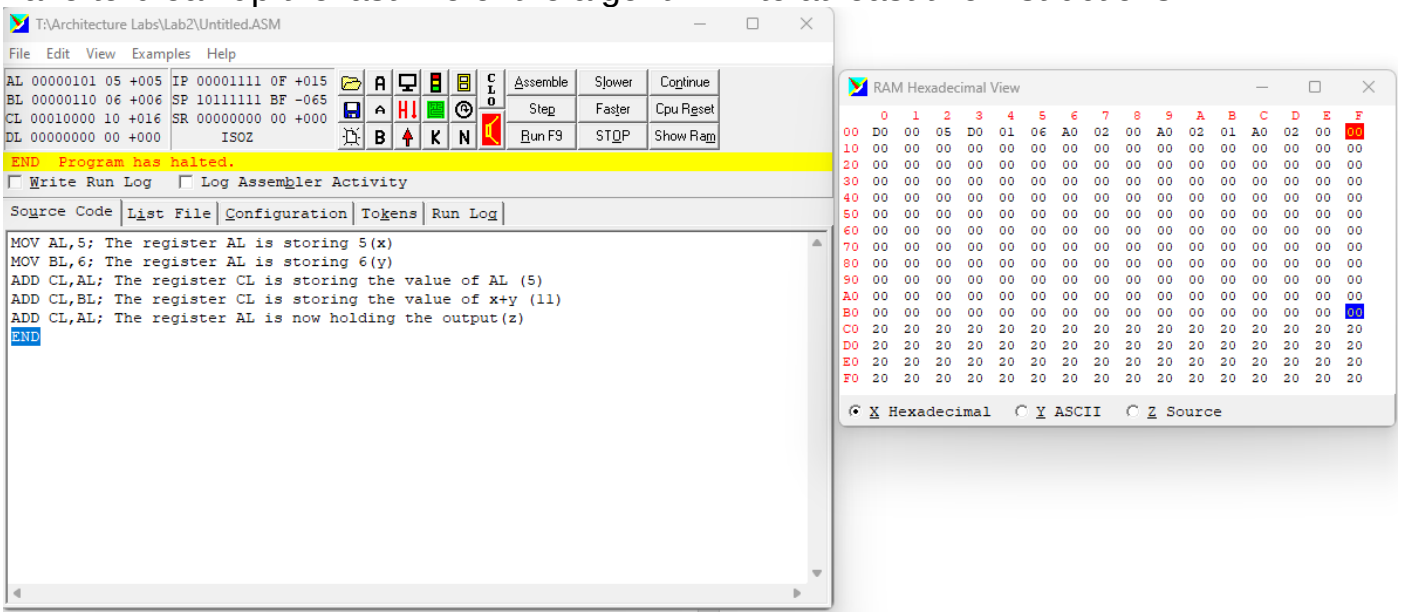
# Exercise:

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

```
x <- 5
y <- 6
z <- x + y + x
```

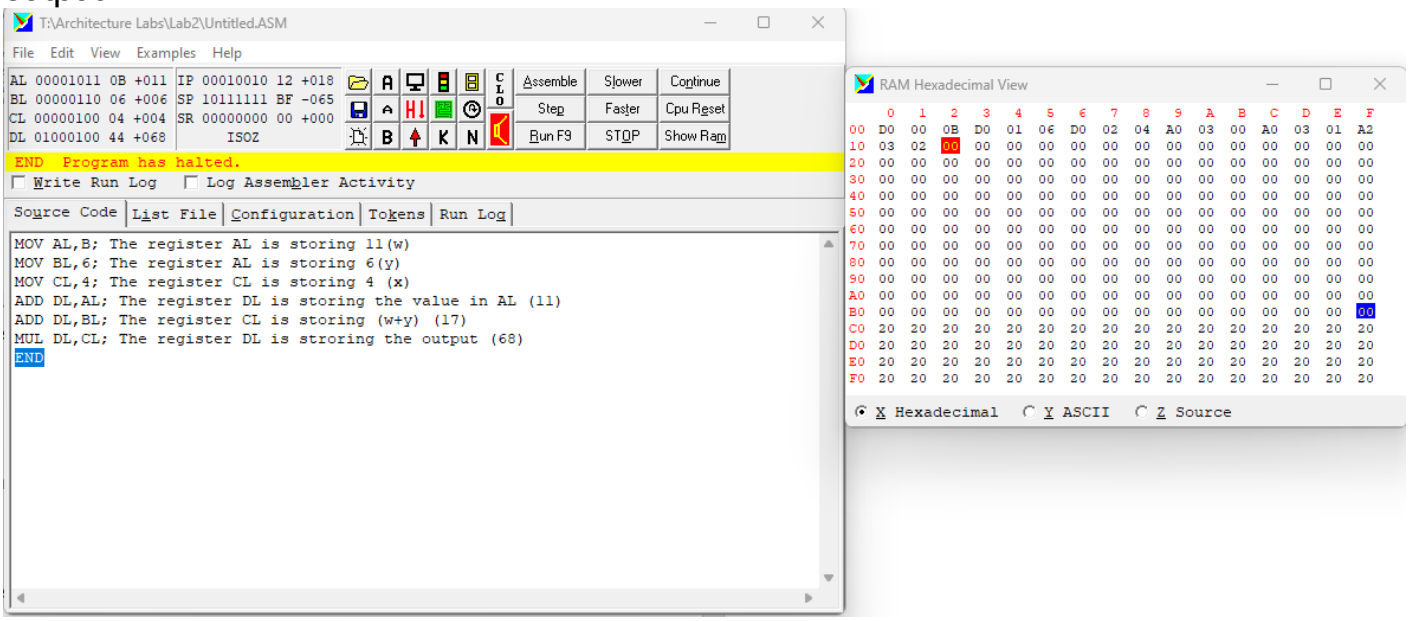
Clearly mark which register you are using to store the output `z` - you may do so with comments in the code or annotations on the screenshots. You will have to break up the last line of the algorithm into at least two instructions.



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

```
w <- 11
y <- 6
x <- 4
z <- (w + y) * x
```

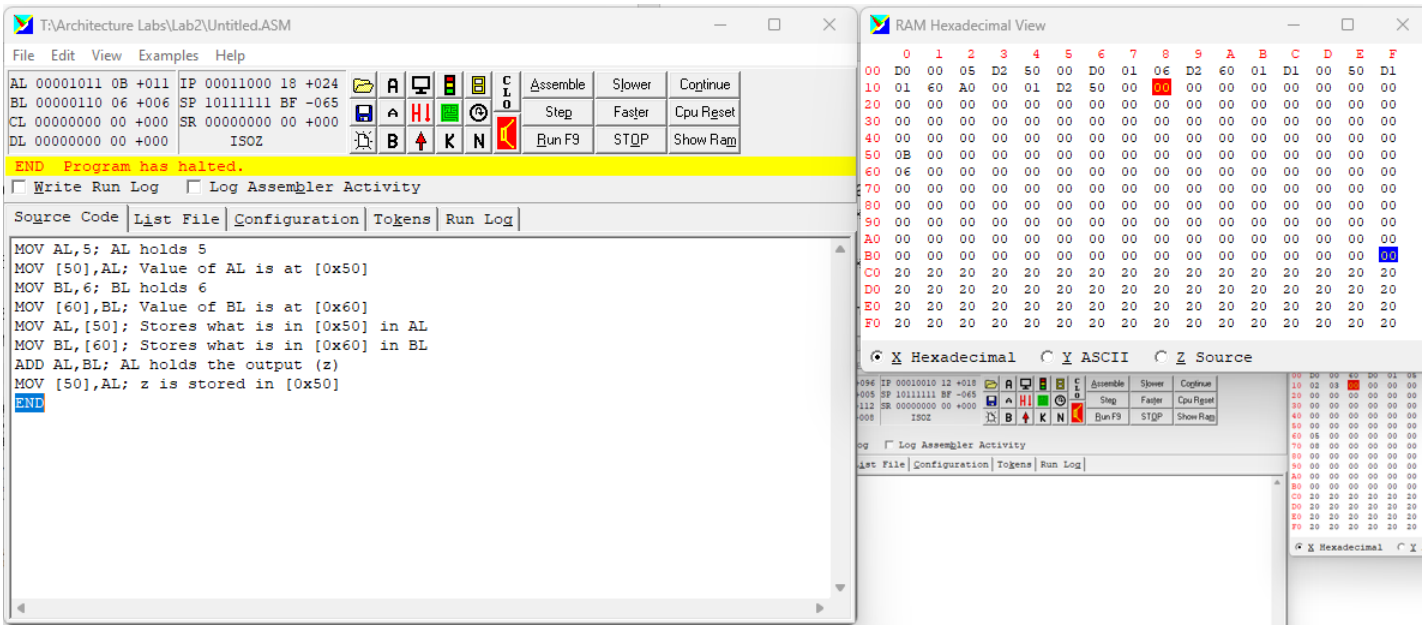
Clearly mark which register you are using to store the output `z` - you may do so with comments in the code or annotations on the screenshots. You will have to convert 11 into hexadecimal, otherwise, you will not get the expected output.



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

```
[0x50] <- 5
[0x60] <- 6
z <- [0x50] + [0x60]
[0x50] <- z
```

Clearly mark which register you are using for the output `z`. Don't take shortcuts: show the instructions for writing to and reading from memory.



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

```
x <- 0x60
[x] <- 5
y <- 0x70
[y] <- 8
```

Use indirect addressing to write to memory.