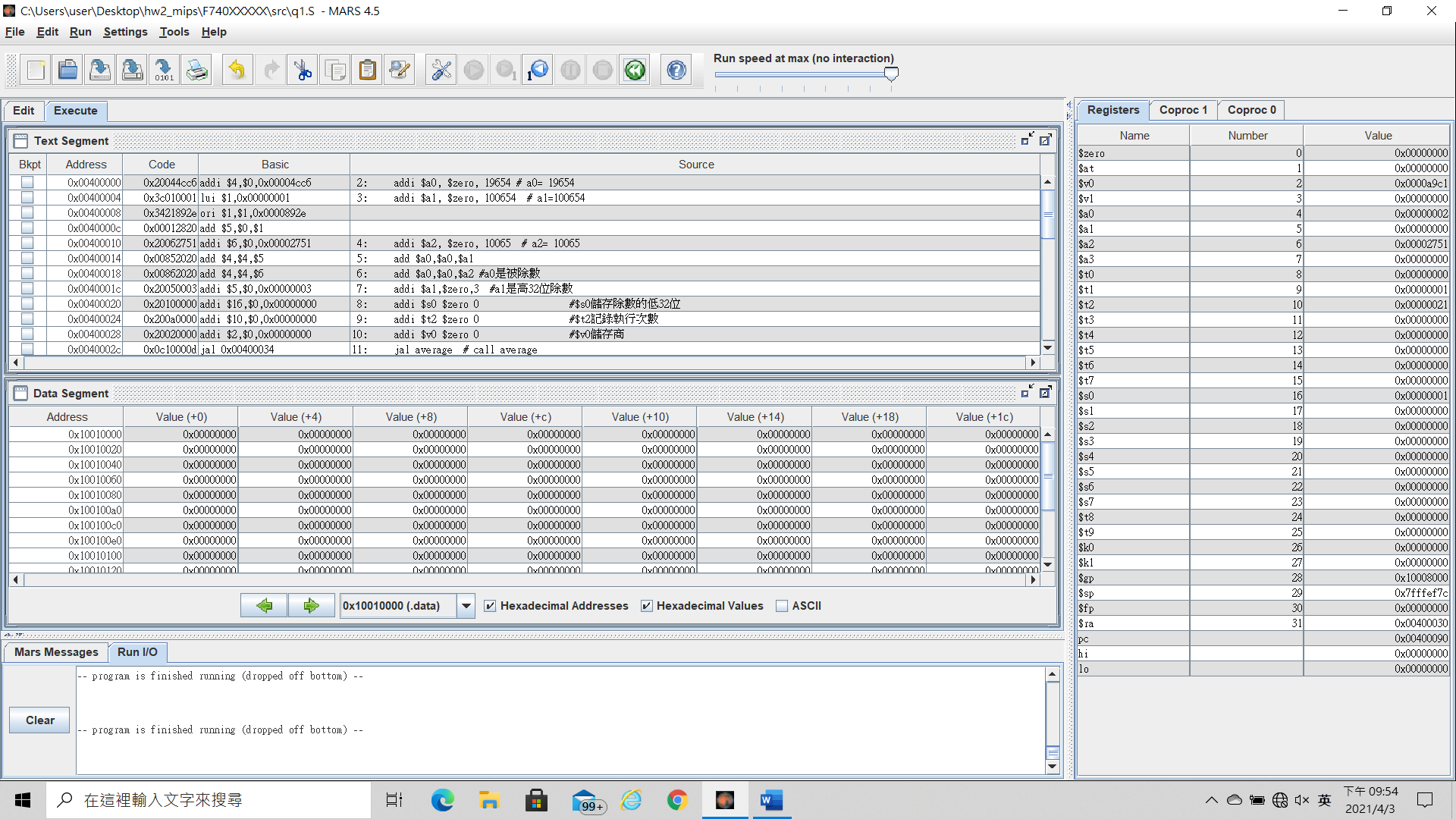
**Computer Organization 2019**

**HOMEWORK 2**

系級: 資訊112 學號: H54084078 姓名: \_\_陳詠君

Program 1 (Find the average)

**實驗結果圖:**



**程式運作流程:**

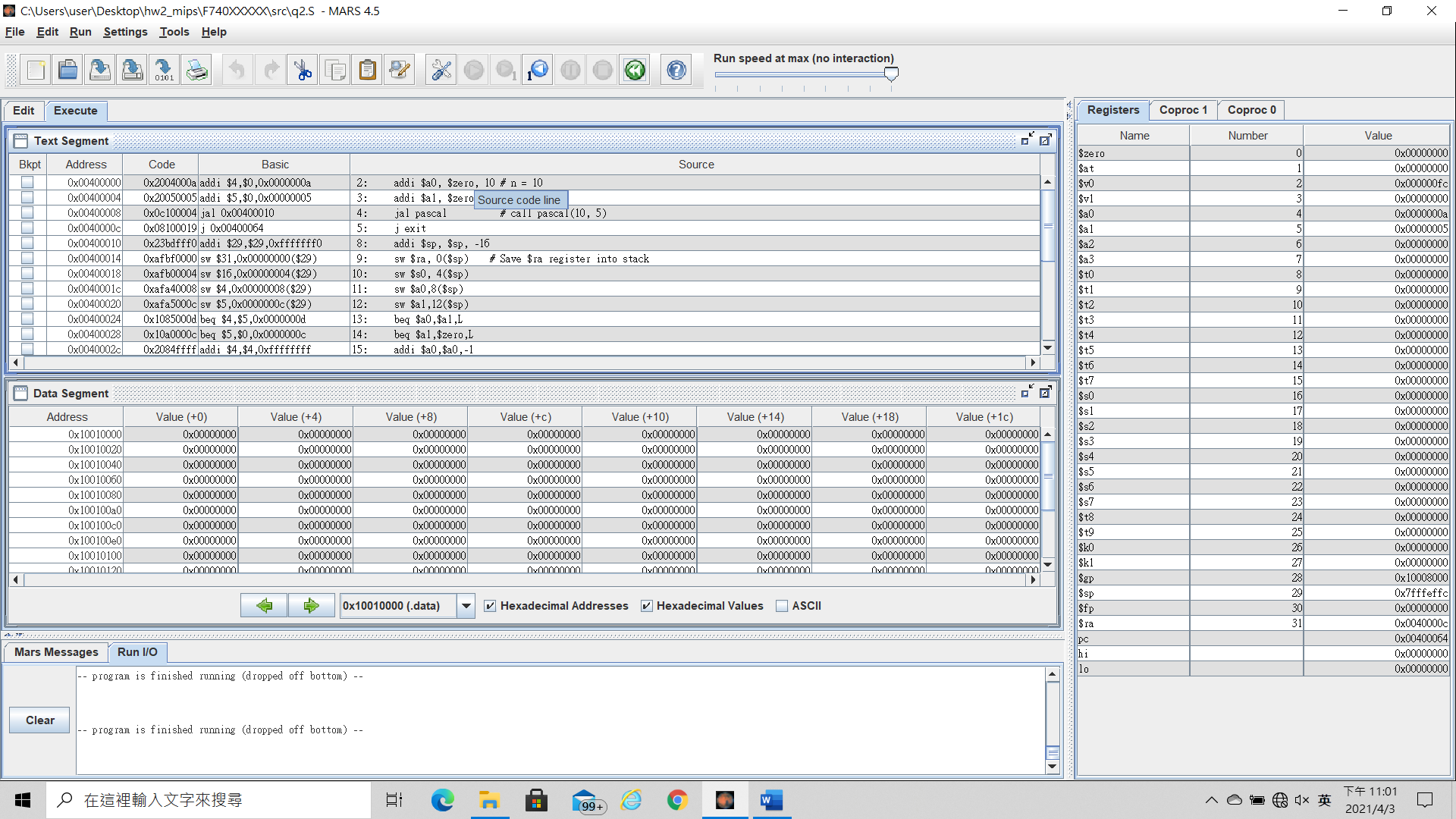
如果被除數大於除數，則商加1 ，被除數=被除數-除數，再重新執行此步驟直到被除數=除數，返回return address。

**心得**

這個作業比我想像中複雜，原本以為用div輕輕鬆鬆可以寫出來，結果看到題目要求說不能用div，我大概想了一天，後來上課老師教到除法器，我才知道原來要用迴圈跟判斷被除數與除數大小，整個寫作業過程一直卡關。不過寫完這次作業後，我更清楚除法器的原理。

Program 2 (Pascal)

**實驗結果圖:**



**程式運作流程:有註解**

main:

addi $a0, $zero, 10 # n = 10

addi $a1, $zero, 5 # m = 5

jal pascal # call pascal(10, 5)

j exit

pascal:

addi $sp, $sp, -16

sw $ra, 0($sp) # Save $ra register into stack

sw $s0, 4($sp)

sw $a0,8($sp)

sw $a1,12($sp)

beq $a0,$a1,L #a0=a1 branch to L

beq $a1,$zero,L #a1=0 branch to L

addi $a0,$a0,-1 #pascal(n-1,m)

jal pascal

add $s0, $zero, $v0 #s0 save value

addi $a1,$a1,-1 #pascal(n-1,m-1)

jal pascal

add $v0, $v0, $s0 #pascal(n-1,m)+pascal(n-1,m-1)

exitpascal:

lw $ra, 0($sp)

lw $s0, 4($sp)

lw $a0, 8($sp)

lw $a1, 12($sp)

addi $sp, $sp, 16

jr $ra #return

L:

addi $v0,$zero,1 #v0=1

j exitpascal

exit:

**心得**

遞迴真的很複雜，很容易搞混return address在哪裡，還有這個作業讓我體會到sw 、lw的重要，如果沒有sw、lw，答案就錯了。沒想到一個簡單的遞迴就可以這麼複雜，原來在簡單的C語言程式背後，是由這寫簡單的指令組合出來的，這次作業讓我更了解記憶體位置，有學到東西。