

109 1 組合語言與嵌入式系統 Final Project

1. 組別及姓名：

組別：35

組員：資訊二乙	10827233	林庭
資訊二乙	10827234	彭桂綺
資訊二乙	10827235	馮信華

2. 程式說明：

程式總共分成3個function：Name, ID, draww, Main。

Name和ID與midterm project都相同，主要在draww這個function，它的主要目的就是賦予Frame二維陣列裡每個元素的值，這些元素的值，會決定它們投影在畫面上的顏色，我們利用老師提供的C範例程式碼，來建立與其具有相同功能的組合語言程式碼，其中有draww, preSet, preFor2, for2, while, whileStatement, processColor, for2Statement, for1Statement等label，下面會重點說明這些label的作用，最終一樣以main呼叫並執行即可。

3. 設計重點說明：

Name:

我們都使用r0來load組別和組員姓名的位置，每load完一個位置就printf印出來，最後便把組別和組員姓名印出完畢。

ID:

我們使用r0來load “%d”，並用r1將學號分別儲存進ALLId這個label，3個學號都輸入並儲存完成之後，用r2, r3, r4來分別儲存學號的位置，再將它們定址，把r2, r3, r4的值add到r5，用r0來load “%s”，並讀取使用者輸入至r1，判斷r1是不是 ‘p’，若是，就把學號和學號的總和分行印出來，若不是，就結束。

以下為draww主要label之說明：

draww:

將fp, lr放入堆疊裡，讓fp指回fp，將r4~r10放入堆疊，r4~r10要來拿儲存某些變數的值，它們在這個function中不會再變成其他變數，在sp建立16bit的空間也是為了儲存變數，賦值給r4~r10，讓它們分別=cX, cY, width=640, height=480, i=0, x=0, y=0為後續的使用，最後呼叫for1Statement這個label。

preFor2:

賦值0給r10，每次for迴圈開始都要將y歸零，呼叫for2Statement這個label。

for2:

r6算術右移1個位元再賦值給r3， $r2=r9-r3$ ，將r1定址為1500拿來做乘法用， $r0=r2*r1$ ，將r3賦值給r1，呼叫__aeabi_idiv來做除法，除完的值會在r0，將r0賦值給r3並存進stack裡，以上可以完成C的 $zx = 1500 * (x - (width>>1)) / (width>>1)$ ； $zy = 1000 * (y - (height>>1)) / (height>>1)$ ；也是相同作法，最後將r8賦值0然後呼叫whileStatement這個label。

while:

load zx的值給r2和r3， $r2=r2*r3$ ，load zy的值給r1和r3， $r3=r1*r3$ ，將它們相減， $r2=r2-r3$ ，將這個值給r0，賦值1000給r1，呼叫__aeabi_idiv來做除法，除完的值會在r0， $r3=r0+r4$ ，將r3存進stack裡，5，讓r3算數左移並賦值給r3，load zy的值給r2， $r0=r2*r3$ ，賦值1000給r1，呼叫__aeabi_idiv來做除法，除完的值會在r0， $r3=r0+r5$ ，將r3存在stack裡，將tmp之值load給zx，最後讓r8--。

whileStatement:

load zx的值給r2和r3， $r2=r2*r3$ ，load zy的值給r1和r3， $r3=r1*r3$ ，將它們相加， $r2=r2+r3$ ，load 399999給r3，比較r2和r3，若 $>$ 則呼叫processColor這個label，若 \leq 則將r8賦值給r3，比較r3和0，若 \geq 則呼叫while這個label。

processColor:

將r8算術左移8個位元再賦值給r3，r8賦值給r2，r2和r3 or在一起變成16bit，將r3儲存進stack裡， $r3=r3$ 的補數並存進stack裡，將r10賦值給r2和r3，r3算術左移2個位元， $r3=r2+r3$ ，r3算術左移8個位元，最後加上陣列的起始位置，再加上x極為目前要存進二維陣列的位置，最後讓r10++。

for2Statement:

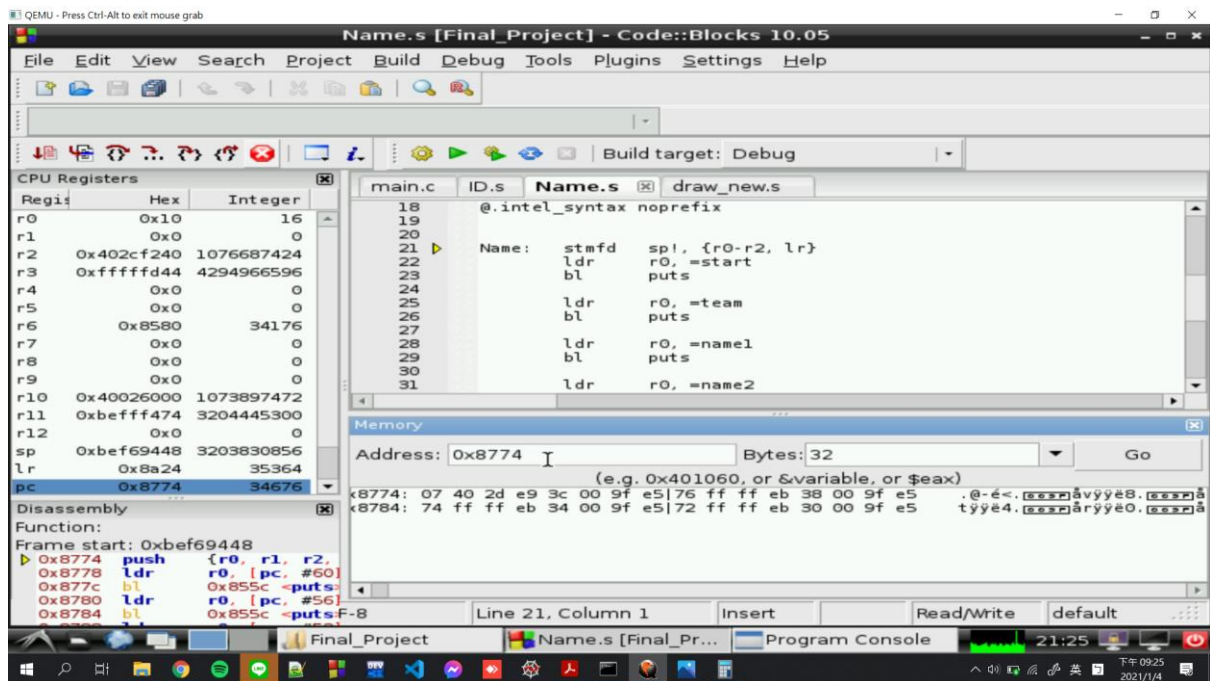
將r10賦值給r2，r7賦值給r3，比較r2和r3之大小，若 $<$ 就進入for2這個label，若 \geq 就讓r9++。

for1Statement:

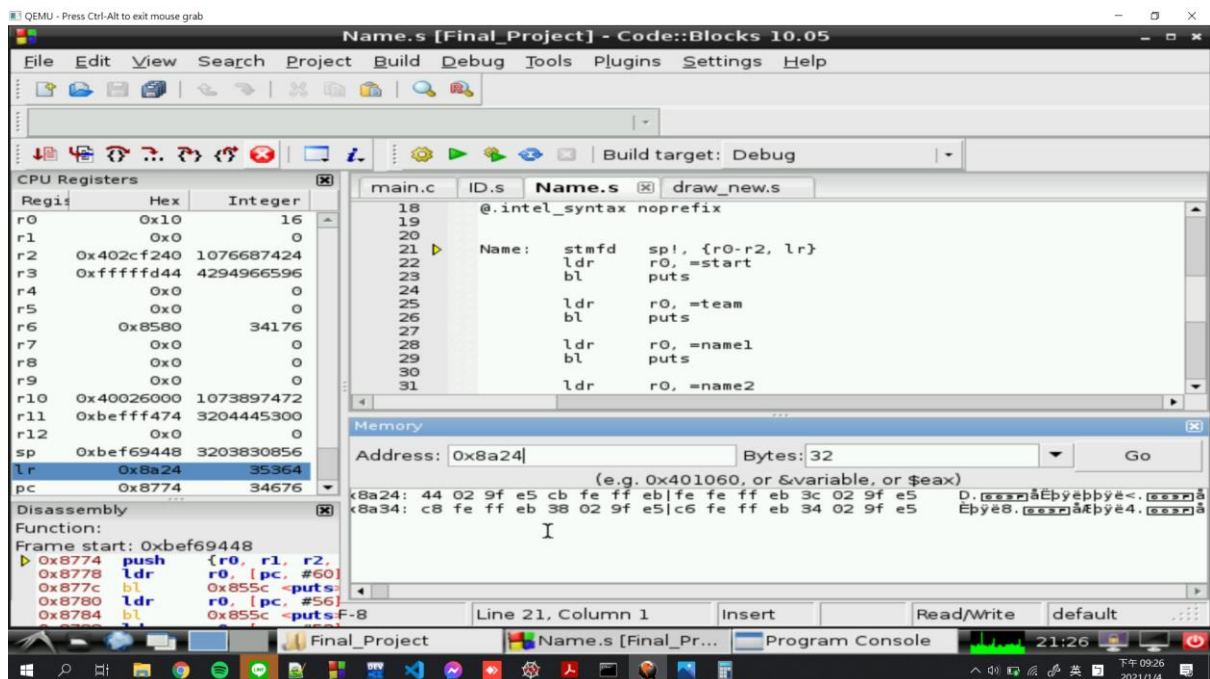
將r9賦值給r2，r6賦值給r3，比較r2和r3之大小，若 \leq 就進入preFor2這個label，若 $>$ 就return並結束。

4. 程式驗證結果、指定之說明項目、相關說明螢幕截圖

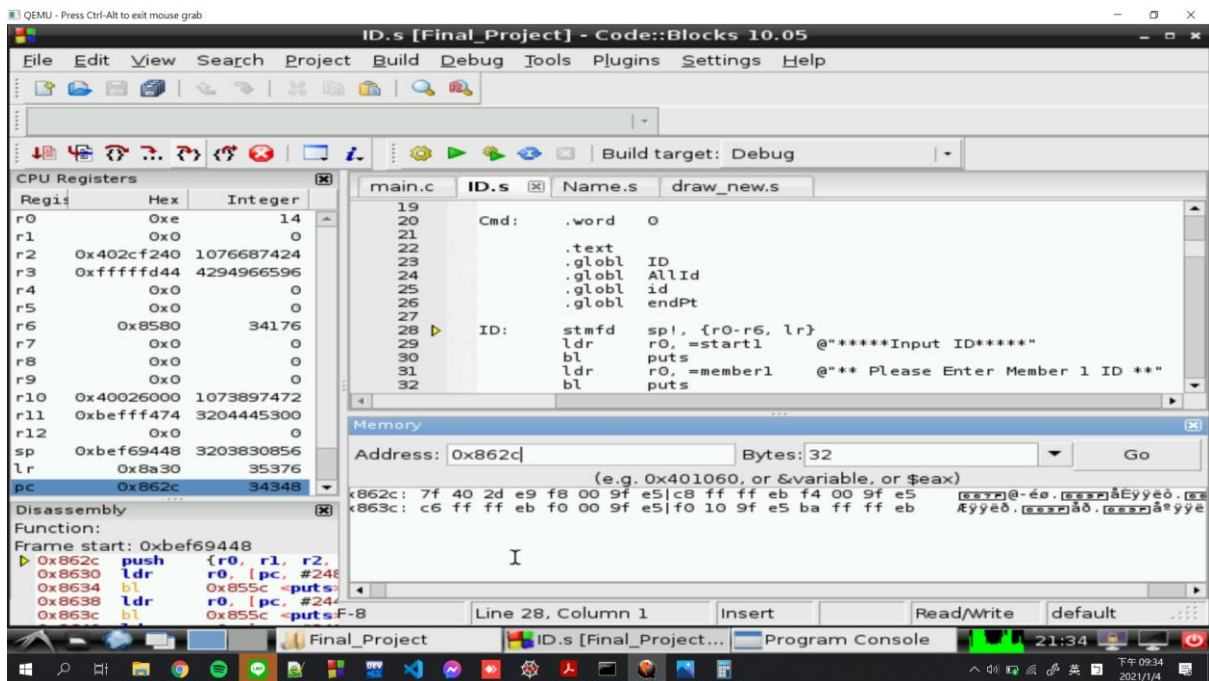
【NAME 起始位址】0x8774



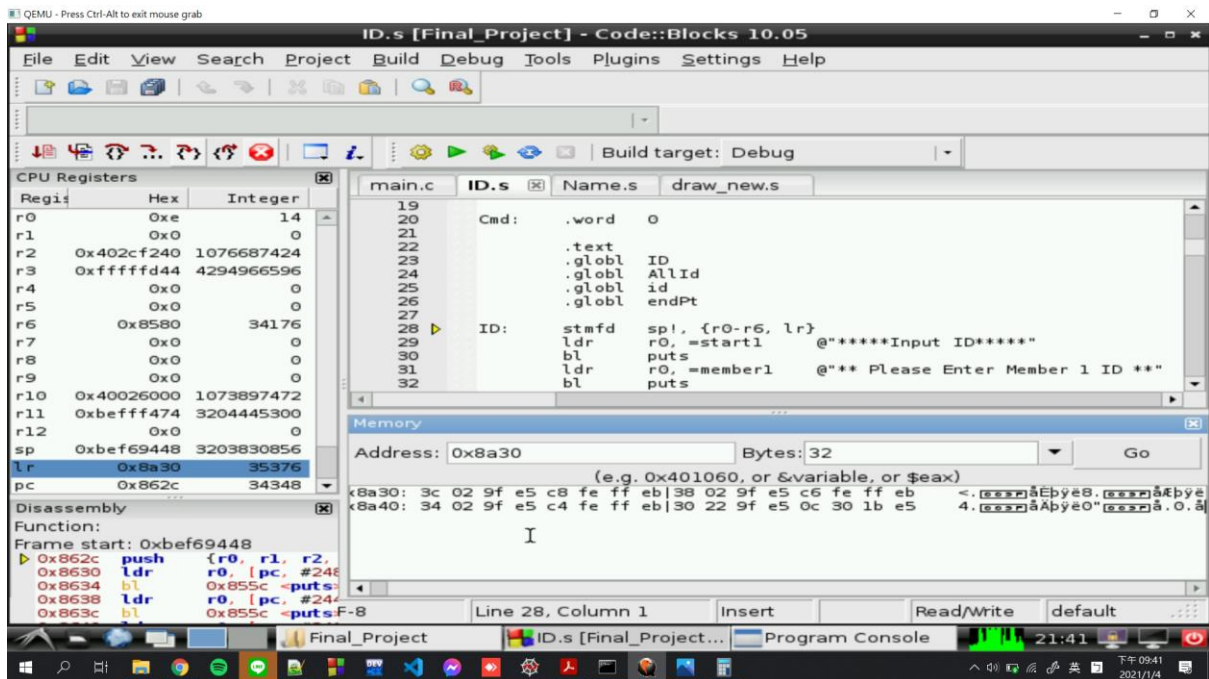
【NAME 結束位址】0x8a24



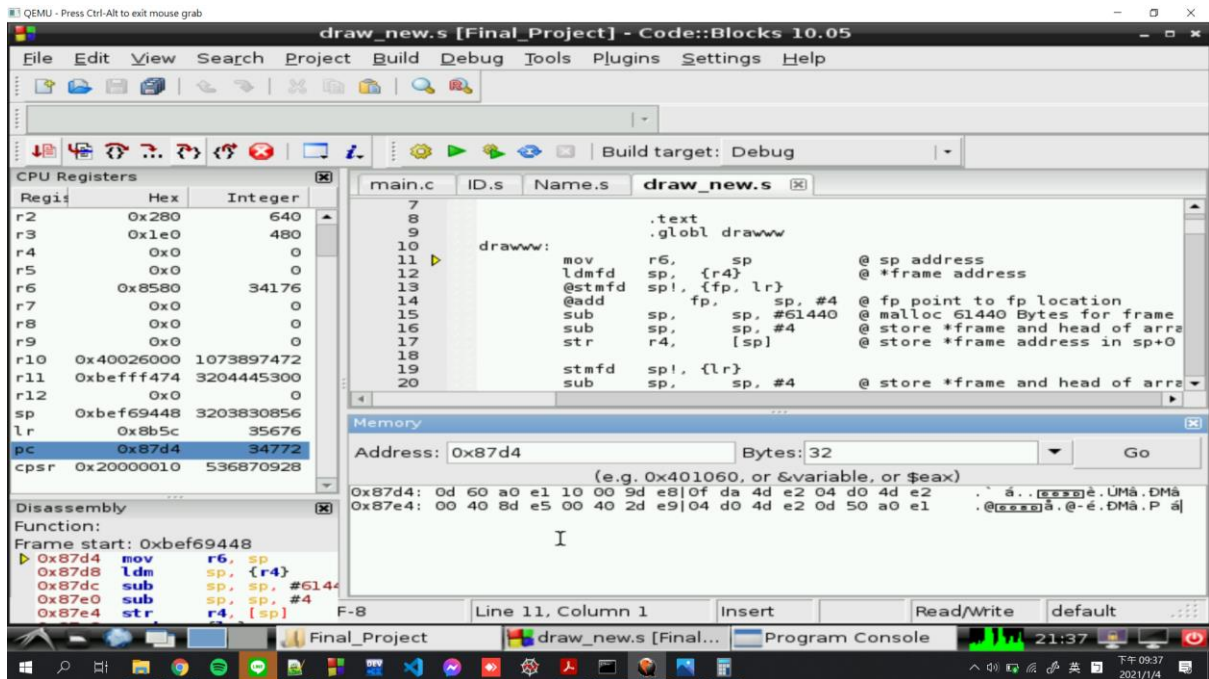
【ID 起始位址】0x862c



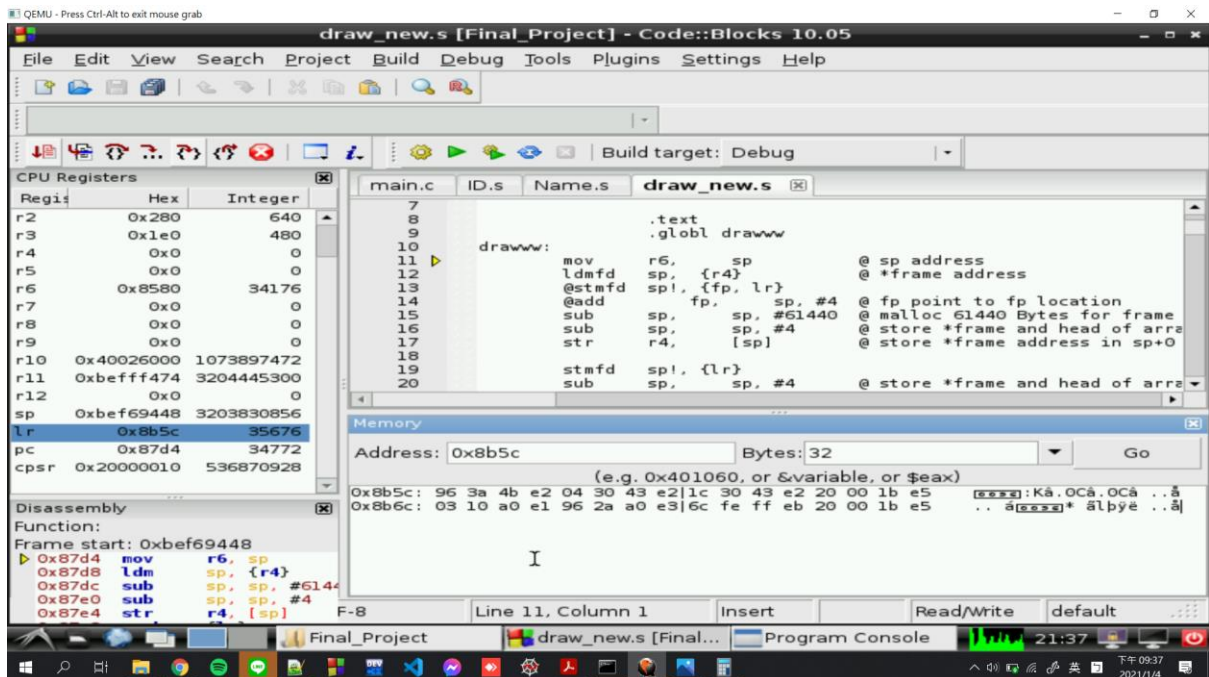
【ID 結束位址】0x8a30



【DRAWWW 起始位址】0x87d4

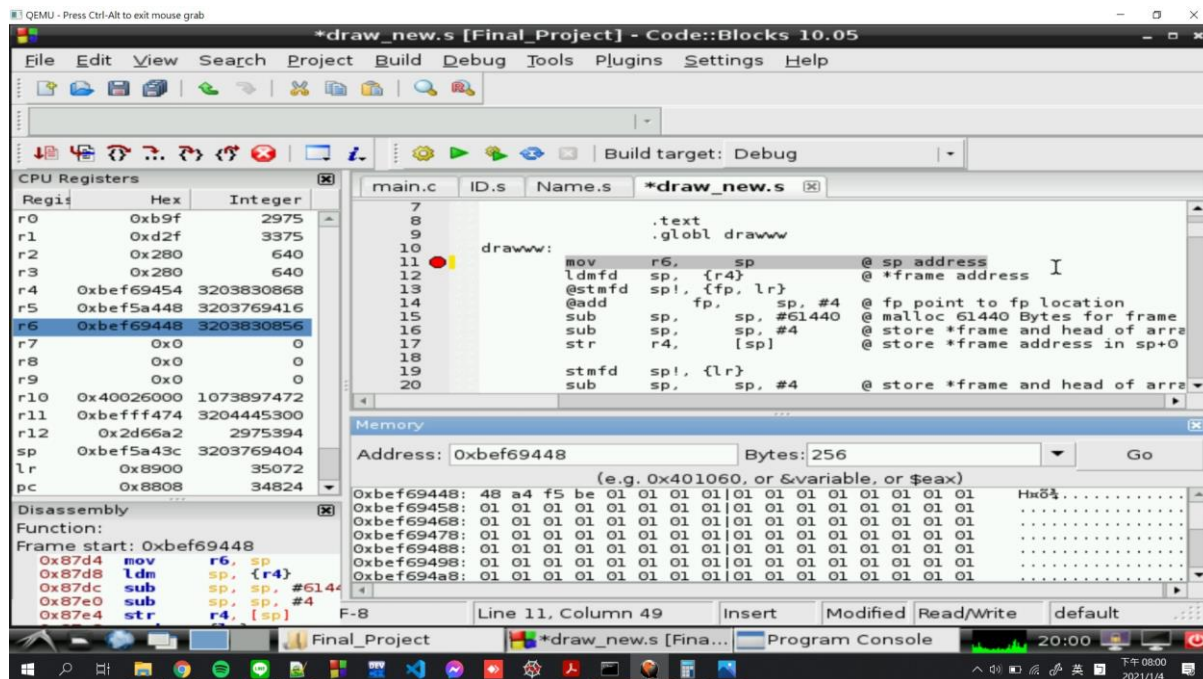


【DRAWWW 結束位址】0x8b5c



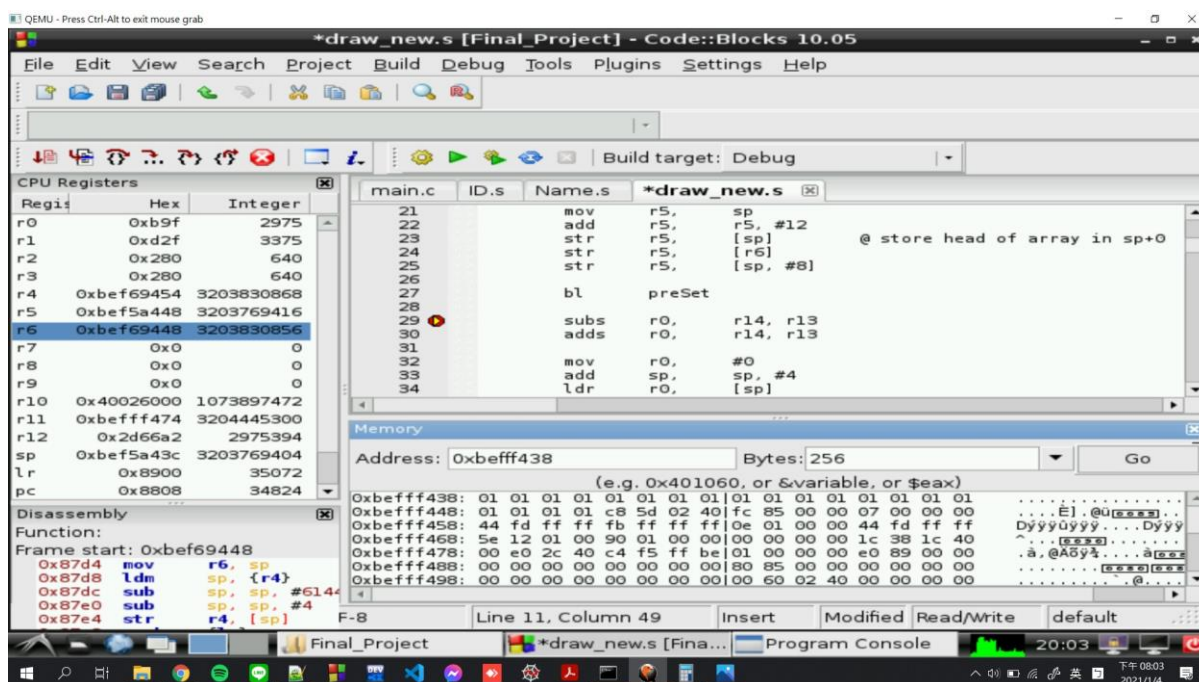
【frame 起始位址】

由 `mov r6, sp` 可知，程式開始時的 `sp` 在 `0xbef69448`，`sp` 目前存 `*frame`，可知 `sp + 4` 即為 `frame` 的起始位址 == `0xbef6944c`。



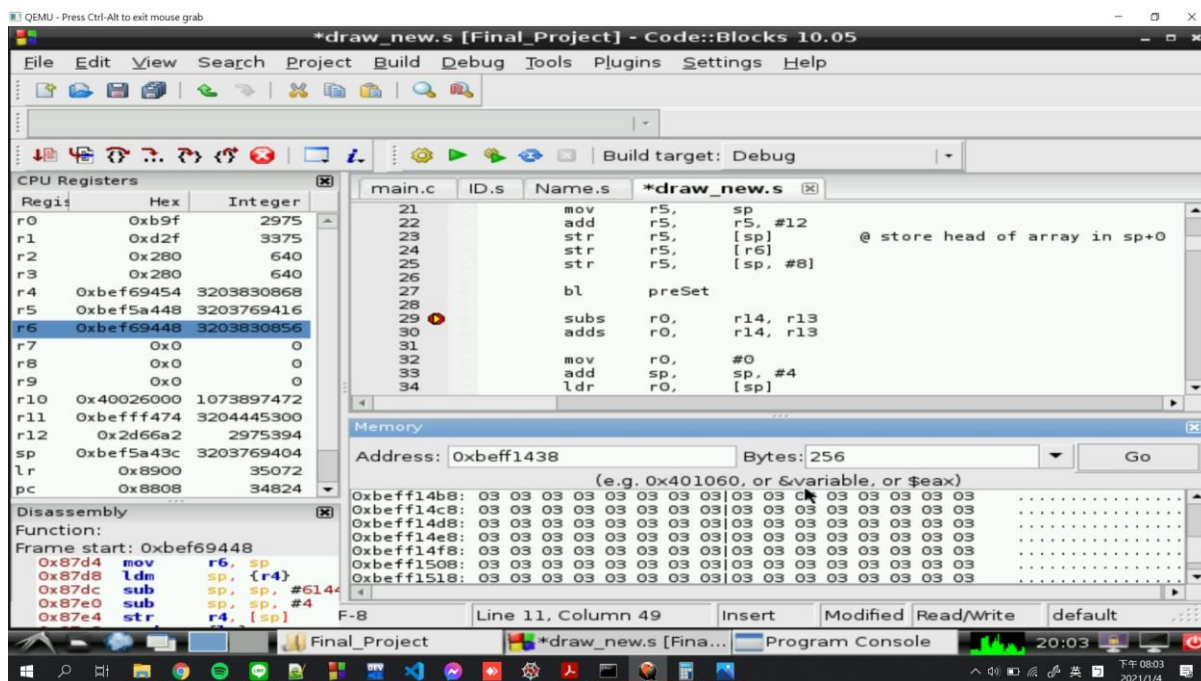
【frame 結束位址】

frame的結束位址在frame 的起始位址 == `0xbef6944c` + 614400 = `0xbefff44c`

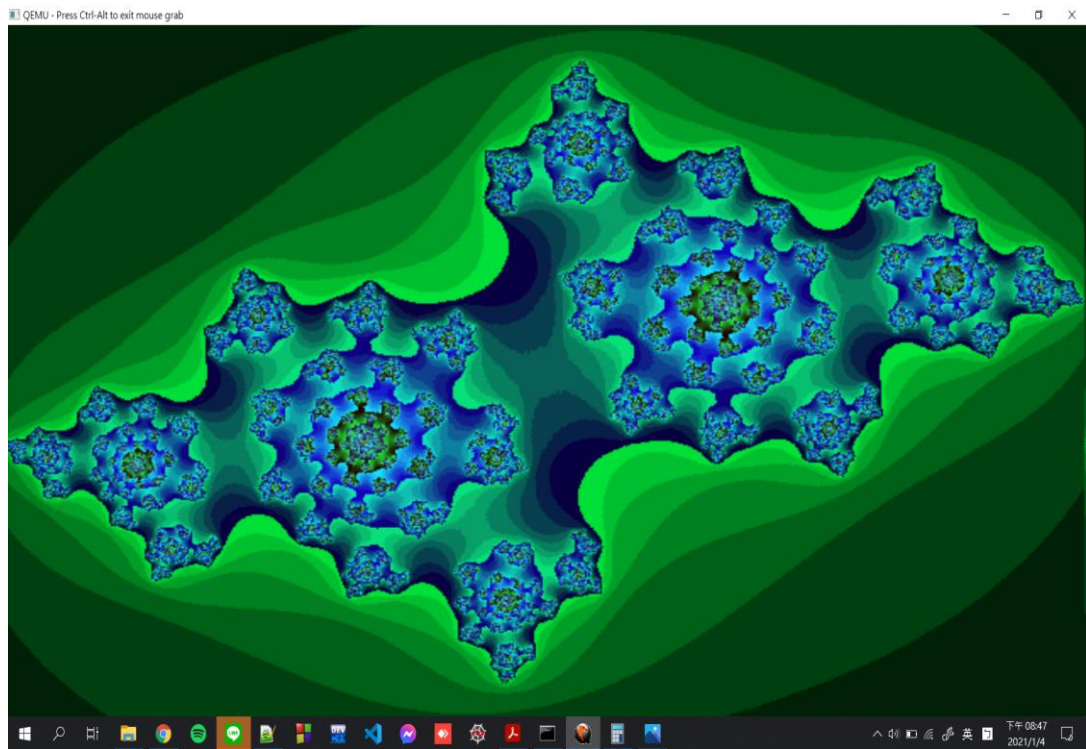


【frame 某段位址】

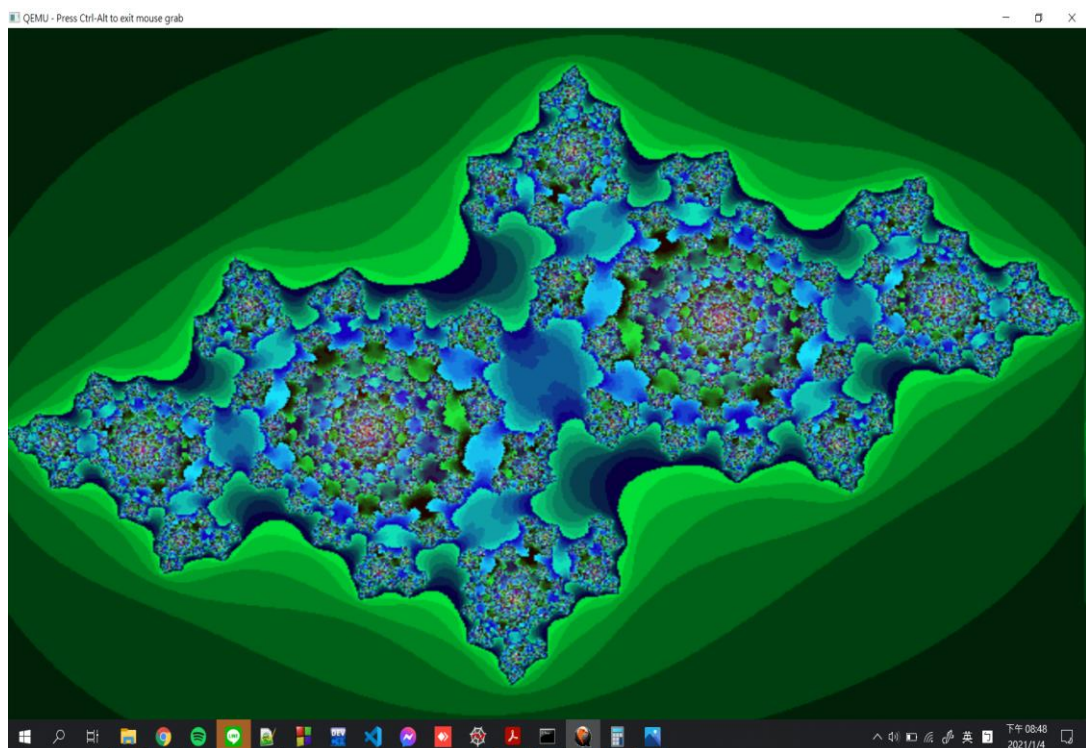
其中記憶體內容存 $303_{(10)} = 771_{(10)}$ 。這個數值代表陣列中某個點的值。這個值會在之後轉會為某個 16-bit 顏色。



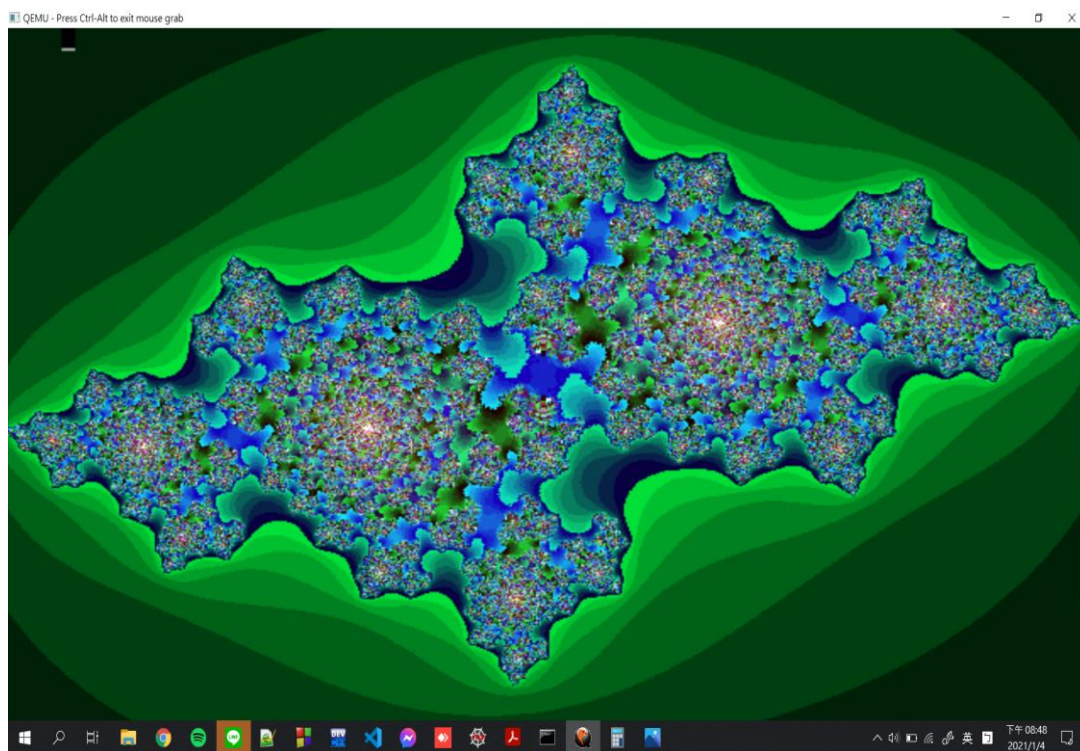
【開始繪製 Julia Set 畫面】



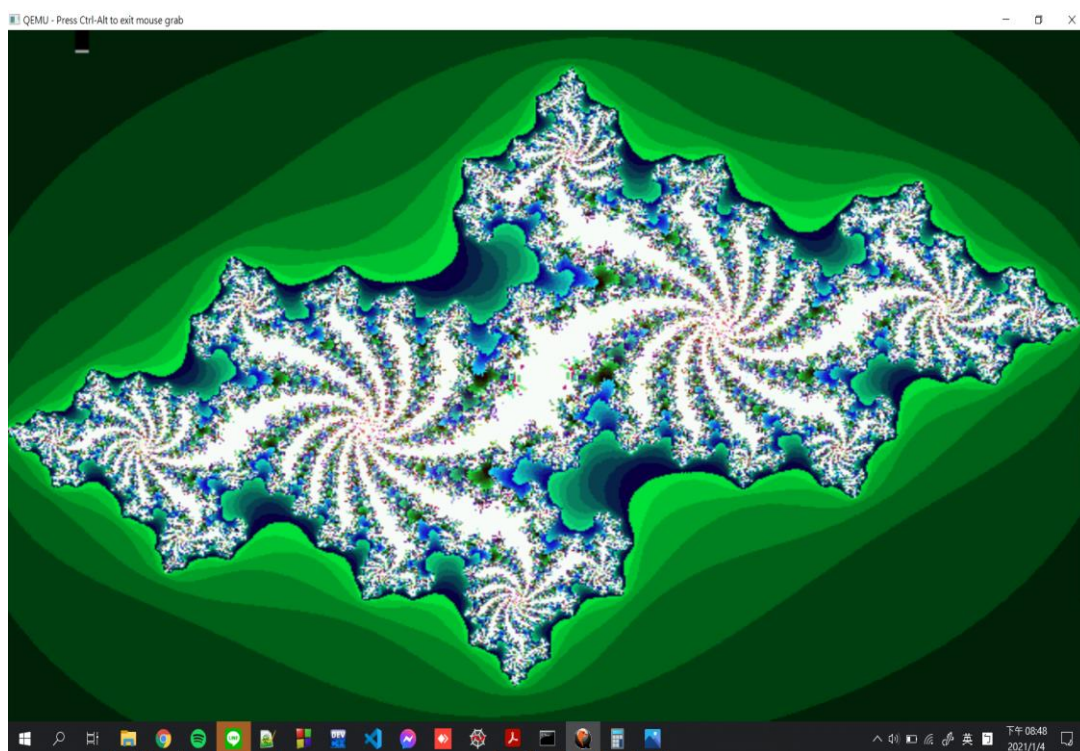
【第 2 張 Julia Set 畫面】



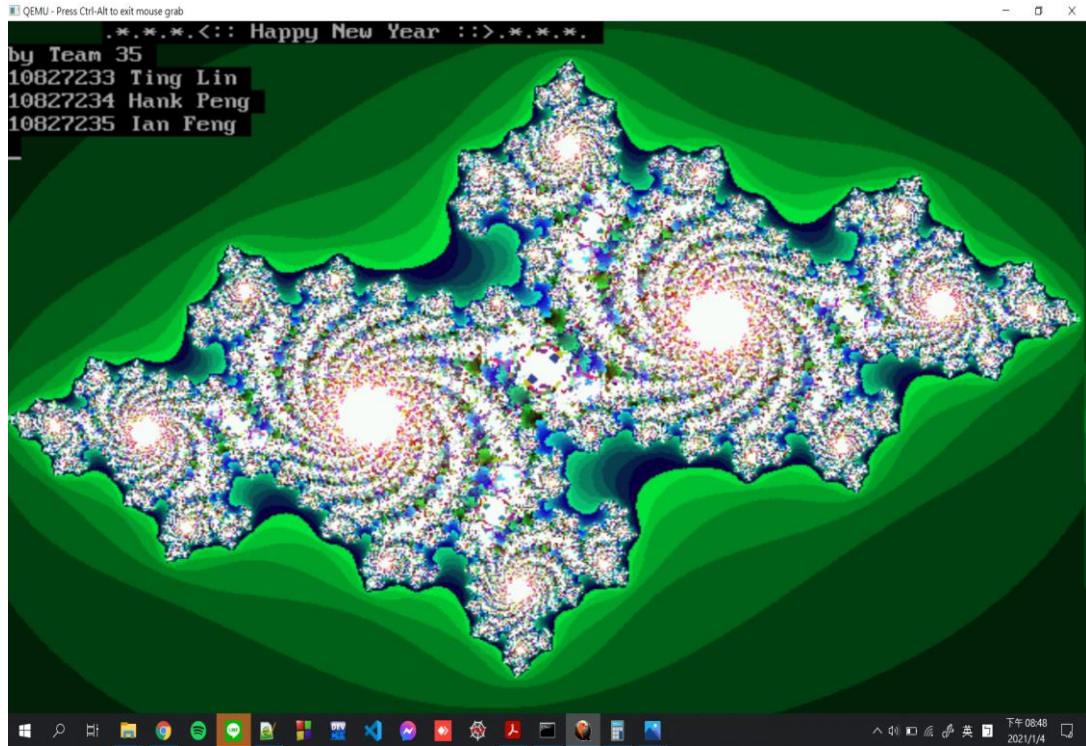
【第 3 張 Julia Set 畫面】



【第 4 張 Julia Set 畫面】



【最後一張 Julia Set 畫面】



5. 心得：

彭桂綺：

期末project的難度真的是期中project無可比擬！一開始真的是無從下手，不知道什麼是碎形，不知道要怎麼用組合語言將數值存進二維陣列...很多的問號。在自己理解跟在組員的討論之後終於了解這次的任務要怎麼下手！原本想說理解要做什麼之後很快就能將程式碼打出來，結果我的想法太天真了！在寫組合語言的時候要了解 stack 中的變數到底是存在哪個位置，光這點我就畫了不知道幾次 stack的配置圖。後來有遇到像是輸出顏色變成藍色，輸出圖片一直往上跑的問題，好險最後都順利解決。做完這Final project真的能夠理解組合語言到底是什麼，也能理解的部分指令要如何運用！也謝謝組員的幫助跟努力付出，我們才能完成這次的任務！期待下學期能再跟組員一組。：)

林庭：

這次project相較於期中難度提升非常多，光是從C的main裡呼叫assembly的function就花了很多時間討論，把drawJuliaset從C轉組語更是一大挑戰，從理解drawJuliaset到安排暫存器與變數存放空間，到轉換程式碼，debug，才發現期中project根本沒什麼難度，只是在為期末暖身罷了。不過其實程式碼轉換反而花比較少的時間，最多是在理解drawJuliaset的參數跟變數意義，做完期末project更加了解組語的實際操作，在這之中更是要特別感謝組員彭桂綺，也發現組語真的不是這麼直觀的語言，希望以後工作上如果有要用到組語的時候我不會全部忘光！：)

馮信華：

這次project難度在於需要把C語言轉換成等價的組合語言程式碼，在理解Julia Set的C就已經花上很多時間了，光C就有些指令沒有看過，在開始打組語程式碼前我們已經討論許久，但是程式碼的部分反而沒花那麼多時間，就一道一道指令慢慢的實現，比較麻煩是暫存器不夠那麼多參數使用，所以要在stack裡建立空間，而這些也使得程式碼變得很難直覺知道他在做什麼，很感謝我的組員們耐心地教導我，我還有許多基本知識沒有很熟，他們還是願意指導我，讓我學到了很多。：)

6. 各組員分工方式與負責項目

程式設計：林庭、彭桂綺、馮信華

程式撰寫：林庭、彭桂綺

程式糾錯：林庭、彭桂綺、馮信華

撰寫報告書=>程式說明：馮信華

程式驗證結果：林庭、彭桂綺