ARM Assembly Language
(Final Project)

108學年度第一學期
指導老師：朱守禮老師
10727214 資工二乙 洪友祥
10727215 資工二乙 沈家丞
10727248 資工二乙 鄭珮慈

一、背景

✦Project1 NAME 函數功能

(1)題目要求：規劃四個記憶體區塊，分別存放組別與組員的英文名字(以上皆已事先填入所屬的記憶體區塊)，並印出組別與組員姓名。除此之外，還必須執行此指令"sbcs r0, r3, r4"。

(2)設計方向：1.事先規劃好記憶體資料
2.載入記憶體位置到暫存器
3.使用printf來輸出

✦Project2 ID 函數功能

(1)題目要求：規劃四個記憶體區塊，分別存取輸入的三個學號和其總和，最後輸入'p'，並輸出四個值。

(2)設計方向：1.讀入資料並存取
2.將資料相加
3.使用printf來輸出

✦Project3 drawJuliaSet 函數功能

(1)題目要求：Operand2使用三種以上、非Branch指令的Condition Execution使用三種以上、還必須執行此指令"adds r14, r0, r1"。

(2)設計方向：依照draw.c檔案的內容逐一將其從c語言翻譯為組合語言

✦Project4 main 函數功能

(1)題目要求：整合前三個函數的功能，先將兩個函數分別執行，最後再將所有的資料全部輸出一遍。

(2)設計方向：1.呼叫前三個函數
2.確認各數值所存放在暫存器的位置
3.使用printf來輸出

二、方法

（只特別說明與期中相異的部分）

✦Project1 NAME 函數

```
2   .data
3
4   title1 :
5     .asciz "*****Print Name*****\n"
6   team :
7     .asciz "Team 52\n"
8   member1 :
9     .asciz "Uxiang Hong\n"
10  member2 :
11    .asciz "Chen Shen\n"
12  member3 :
13    .asciz "Angela Cheng\n"
14  end1 :
15    .asciz "*****End Print*****\n"
16
17  .text
18  .global NAME
19
```

```
21  NAME :
22    stmfd sp!, {r4-r12, lr}
23    mov   r4, r0
24    mov   r5, r1
25    mov   r6, r2
26    mov   r7, r3
27    ldr   r0, =title1
28    bl    printf
29
30    ldr   r0, =team
31    bl    printf
32    mov   r0, r4
33    ldr   r1, =team
34    bl    strcpy
35
36    ldr   r0, =member1
37    bl    printf
38    mov   r0, r5
39    ldr   r1, =member1
40    bl    strcpy
41
42    ldr   r0, =member2
43    bl    printf
44    mov   r0, r6
45    ldr   r1, =member2
46    bl    strcpy
47
48    ldr   r0, =member3
49    bl    printf
50    mov   r0, r7
51    ldr   r1, =member3
52    bl    strcpy
```

22. 備份r4-r12跟lr的位置到sp
23-26.備份r0-r3到r4-r7
中間做Printf動作
strcpy是將r1複製到r0存到該有的暫存器

```
53
54    ldr   r0, =end1
55    bl    printf
56
57    sbcs  r0, r3, r4
58    mov r0, #0
59    ldmfd sp!, {r4-r12,pc}
60
61  .end
```

57.r0=r3-r4

58.將r0清空

59.還原r4-r12跟lr的位置從sp

```
.data

Enter :
  .asciz "\n"
title2 :
  .asciz "*****Input ID*****\n"
sum :
  .word 0
id1 :
  .word 0
id2 :
  .word 0
id3 :
  .word 0
inputID :
  .asciz "%d"
str1 :
  .asciz "** Please Enter Member 1 ID:**\n"
str2 :
  .asciz "** Please Enter Member 2 ID:**\n"
str3 :
  .asciz "** Please Enter Member 3 ID:**\n"
str4 :
  .asciz "** Please Enter Command **\n"
cmd :
  .asciz " "
cmds :
  .asciz "%s"
commandP :
  .byte 'p'
info1 :
  .asciz "*****Print Team Member ID and ID Summation*****\n"
info2 :
  .asciz "ID Summation = "
end2 :
  .asciz "\n*****End Print*****\n"

.text
.global ID
```

✦Project2 ID 函數

```
41  ID :
42    stmfd sp!, {r4-r12, lr}
43    mov   r5, r0
44    mov   r6, r1
45    mov   r7, r2
46    mov   r8, r3
47    ldr   r0, =title2
48    bl    printf
49
50    ldr   r0, =str1
51    bl    printf
52    ldr   r0, =inputID
53    ldr   r1, =id1
54    bl    scanf
55
56    ldr   r0, =str2
57    bl    printf
58    ldr   r0, =inputID
59    ldr   r1, =id2
60    bl    scanf
61
62    ldr   r0, =str3
63    bl    printf
64    ldr   r0, =inputID
65    ldr   r1, =id3
66    bl    scanf
67
68    ldr   r0, =str4
69    bl    printf
70
71    ldr   r0, =cmds
72    ldr   r1, =cmd
73    bl    scanf
74    ldr   r0, =commandP
75    ldr   r1, =cmd
76    ldrb  r0, [r0]
77    ldrb  r2, [r1]
78    cmp   r2, r0
```

42.備份r4-r12跟lr到sp
43~46將r0-r3 copy 到r5-r8

```
    ldreq r0, =info1
    bleq  printf

    ldr   r0, =inputID
    ldr   r1, =id1
    ldreq r1,[r1]
    str   r1, [r5]
    ldrne r1,[r1]
    moveq r9, r1
    moveq r9, r1, lsl #0
    moveq r9, r1, lsr #0
    bleq  printf
    ldreq r0, =Enter
    bleq  printf

    ldr   r0, =inputID
    ldr   r1, =id2
    ldreq r1,[r1]
    str   r1, [r6]
    moveq r10, r1
    bleq  printf
    ldreq r0, =Enter
    bleq  printf

    ldr   r0, =inputID
    ldr   r1, =id3
    ldreq r1,[r1]
    str   r1, [r7]
    moveq r11, r1
    bleq  printf
    ldreq r0, =Enter
    bleq  printf
    ldreq r0, =Enter
    bleq  printf
```

```
115     ldreq r0, =id1
116     ldreq r0, [r0]
117     ldreq r1, =id2
118     ldreq r1, [r1], r1
119     ldreq r2, =id3
120     ldreq r2, [r2], #2
121     mov   r3, #0
122
123     addvc r3, r0, r1
124     addvc r3, r3, r2
125     ldr   r4, =sum
126     str   r3, [r4]
127
128     ldreq r0, =info2
129     bleq  printf
130     ldreq r0, =inputID
131     ldr   r1, =sum
132     moveq r4, r1
133     ldreq r4,[r4]
134     str   r4, [r8]
135     ldreq r1,[r1]
136     bleq  printf
137     ldreq r0, =end2
138     bleq  printf
139
140     mov   r0, r5
141     mov   r1, r6
142     mov   r2, r7
143     mov   r3, r8
144     ldmfd sp!, {r4-r12, pc}
145
146   .end
```

140-143.將r5-r8複製回r0-r3

144.還原sp至r4-r12跟pc

✦Project3 drawJuliaSet 函數

```
    .global __aeabi_idiv
    .text
    .global drawJuliaSet
drawJuliaSet:
    stmfd   sp!, {r4, lr}
    add r4, sp, #4
    sub sp, sp, #48
    str r0, [r4, #-40] @cX
    str r1, [r4, #-44] @cY
    str r2, [r4, #-48] @width
    str r3, [r4, #-52] @height
    mov r3, #255
    str r3, [r4, #-28] @maxIter
    mov r3, #0  @ x = 0
    str r3, [r4, #-20] @x
    b   Big_For_Judge
Big_For_Assign:
    mov r3, #0
    str r3, [r4, #-24] @y
    b   Small_For_Judge
```

```
21  Small_For_Working:
22      ldr r3, [r4, #-48]
23      mov r3, r3, asr #1
24      ldr r2, [r4, #-20]
25      rsb r3, r3, r2
26      ldr r2, computeNum
27      mul r2, r2, r3
28      ldr r3, [r4, #-48]
29      mov r3, r3, asr #1
30      mov r0, r2
31      mov r1, r3
32      bl  __aeabi_idiv
33      mov r3, r0
34      str r3, [r4, #-8] @zx
35
36      ldr r3, [r4, #-52]
37      mov r3, r3, asr #1 @height>>1
38      ldr r2, [r4, #-24]
39      rsb r3, r3, r2
40      mov r2, #1000
41      mul r2, r2, r3
42      ldr r3, [r4, #-52]
43      mov r3, r3, asr #1
44      mov r0, r2
45      mov r1, r3
46      bl  __aeabi_idiv
47      mov r3, r0
48      str r3, [r4, #-12] @zy
49
50      ldr r3, [r4, #-28]
51      str r3, [r4, #-16] @i
52      b   While_Judge
```

22~34. zx = 1500 * (x - (width>>1)) / (width>>1)

36~48. zy = 1000 * (y - (height>>1)) / (height>>1)

50~52. i = maxIter

```
53  While:
54      ldr r3, [r4, #-8]
55      mov r2, r3
56      mul r2, r2, r3 @ zx * zx
57      ldr r3, [r4, #-12]
58      mov r1, r3
59      mul r3, r1, r3 @ zy * zy
60      rsb r3, r3, r2 @ zx * zx - zy * zy
61
62      ldr r2, computeNum+4
63      smull   r1, r2, r2, r3
64      mov r2, r2, asr #6
65      mov r3, r3, asr #31
66      rsb r2, r3, r2
67
68      ldr r3, [r4, #-40] @cX
69      add r3, r2, r3
70      str r3, [r4, #-32] @tmp
```

63. smull 32位元的乘法

```
72      ldr r3, [r4, #-8]
73      mov r3, r3, lsl #1 @ 2 * zx
74      mov r3, r3, lsr #0
75      ldr r2, [r4, #-12]
76      mul r3, r2, r3 @ 2 * zx * zy
77
78      ldr r2, computeNum+4
79      smull   r1, r2, r2, r3
80      mov r2, r2, asr #6
81      mov r3, r3, asr #31
82      rsb r2, r3, r2
83
84      ldr r3, [r4, #-44] @ cY
85      add r3, r2, r3
86      str r3, [r4, #-12]
87
88      ldr r3, [r4, #-32]
89      str r3, [r4, #-8] @ zx = tmp
90
91      ldr r3, [r4, #-16]
92      sub r3, r3, #1 @i--
93      str r3, [r4, #-16]
94  While_Judge:
95      ldr r3, [r4, #-8]
96      mov r2,r3
97      mul r2, r2, r3 @ zx * zx
98      ldr r3, [r4, #-12]
99      mov r1, r3
100     mul r3, r1, r3 @ zy * zy
101     add r2, r2, r3
102     ldr r3, computeNum+8 @3999999
103     cmp r2, r3
104     bge ColorCountrol @ zx * zx + zy * zy < 4000000 (>=4000000)
105     ldr r3, [r4, #-16]
106     cmp r3, #0
107     bgt While @ i > 0
```

```
108   ColorCountrol:
109       ldr r3, [r4, #-16] @i
110       mov r3, r3, asl #8
111       uxth    r2, r3
112       ldr r3, [r4, #-16]
113       orr r3, r2, r3 @ ((i&0xff)<<8) | (i&0xff)
114       strh    r3, [r4, #-34]
115       ldrh    r3, [r4, #-34]
116       mvn r3, r3 @ r3 => -r3
117       strh    r3, [r4, #-34]
118
119       ldr r2, [r4, #-24]  @y
120       mov r3, r2
121       mov r3, r3, asl #2
122       add r3, r3, r2
123       mov r3, r3, asl #8  @640*2
124       ldr r2, [r4, #4]
125       add r2, r2, r3
126       ldr r3, [r4, #-20]  @x
127       mov r3, r3, asl #1
128       add r3, r2, r3
129
130       ldrh    r2, [r4, #-34]
131       strh    r2, [r3, #0]
132
133       ldr r3, [r4, #-24]
134       add r3, r3, #1 @ y++
135       str r3, [r4, #-24]
```

119~131. frame[y][x] = color

```
136 Small_For_Judge:
137     ldr r2, [r4, #-24]
138     ldr r3, [r4, #-52]
139     cmp r2, r3
140     blt Small_For_Working
141     ldr r3, [r4, #-20]
142     add r3, r3, #1
143     str r3, [r4, #-20]
144 Big_For_Judge:
145     ldr r2, [r4, #-20]
146     ldr r3, [r4, #-48]
147     cmp r2, r3
148     blt Big_For_Assign
149     sub sp, r4, #4
150     adds r14, r0, r1
151     ldmfd   sp!, {r4, pc}
152 computeNum:
153     .word   1500
154     .word   274877907
155     .word   40000000
```

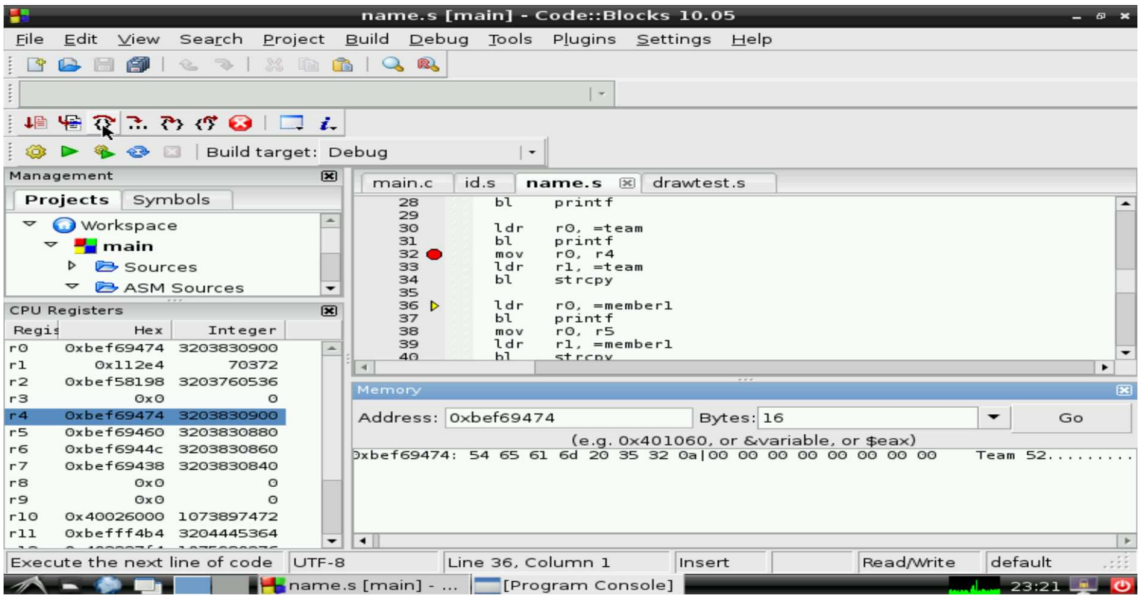
✦Project4 main 函數

```
int main()
{
    //RGB16
    int16_t frame[FRAME_HEIGHT][FRAME_WIDTH];

    int max_cX = -700;
    int min_cY = 270;

    int cY_step = -5;
    int cX = -700;  // x = -700~-700
    int cY;         // y = 400~270

    int fd;

    char team[20] ;
    char name1[20] ;
    char name2[20] ;
    char name3[20] ;

    int *id1 = malloc(sizeof(int));
    int *id2 = malloc(sizeof(int));
    int *id3 = malloc(sizeof(int));
    int *sum = malloc(sizeof(int));

    printf( "Function1: Name\n" );

    //Dummy Function. Please refer to the specification of Project 1.
    NAME(team, name1, name2, name3 ) ;

    printf( "Function2: ID\n" );

    //Dummy Function. Please refer to the specification of Project 1.
    ID(id1, id2, id3, sum);

    //Dummy printout. Please refer to the specification of Project 1.
    printf( "Main Function:\n" );
    printf( "*****Print All*****\n" );
    printf("%s\n", team ) ;
    printf("%d  %s\n",*id1,name1);
    printf("%d  %s\n",*id2,name2);
    printf("%d  %s\n",*id3,name3);
    printf( "ID Summation = %d\n",*sum );
    printf( "*****End Print*****\n" );


    printf( "\n***** Please enter p to draw Julia Set animation *****\n" );

```

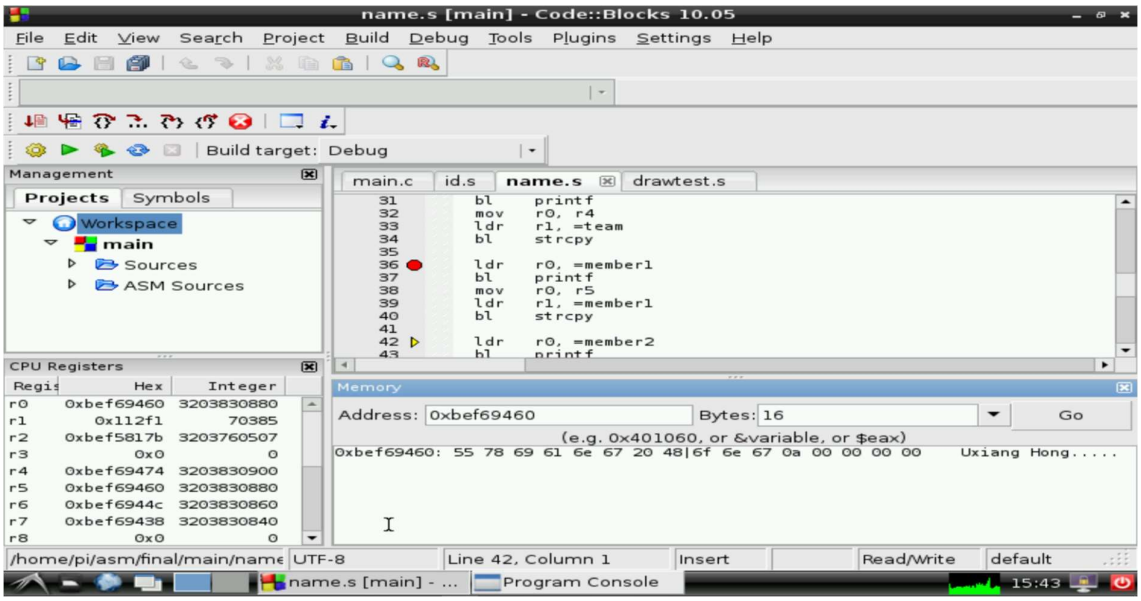
```
77      while(getchar()!='p') {}
78
79      // 清除畫面
80      system( "clear" );
81
82      // 打開 Frame Buffer 硬體裝置的Device Node，準備之後的驅動程式呼叫
83      fd = open( FRAME_BUFFER_DEVICE, (O_RDWR | O_SYNC) );
84
85      if( fd<0 )
86      {   printf( "Frame Buffer Device Open Error!!\n" ); }
87      else
88      {
89          for( cY=400 ; cY>=min_cY; cY = cY + cY_step ) {
90
91              // 計算目前cX,cY參數下的Julia set畫面
92              drawJuliaSet( cX, cY, FRAME_WIDTH, FRAME_HEIGHT, frame );
93
94              // 透過低階I/O操作呼叫Frame Buffer的驅動程式
95              // (將畫面資料寫入Frame Buffer)
96              write( fd, frame, sizeof(int16_t)*FRAME_HEIGHT*FRAME_WIDTH );
97
98              // 移動檔案操作位置至最前端，以便下一次的畫面重新寫入
99              lseek( fd, 0, SEEK_SET );
00          }
105             printf( ".*.*.*.<:: Happy New Year ::>.*.*.*.\n" );
106
107             printf("by %s\n", team ) ;
108             printf("%d  %s\n",*id1,name1);
109             printf("%d  %s\n",*id2,name2);
110             printf("%d  %s\n",*id3,name3);
111
112             // 關閉 Device Node檔案，結束驅動程式的使用
113             close( fd );
114         }
115
116         // 等待使用者輸入正確指令
117         while(getchar()!='p') {
118         }
119
120         return 0;
121     }
```

三、結果

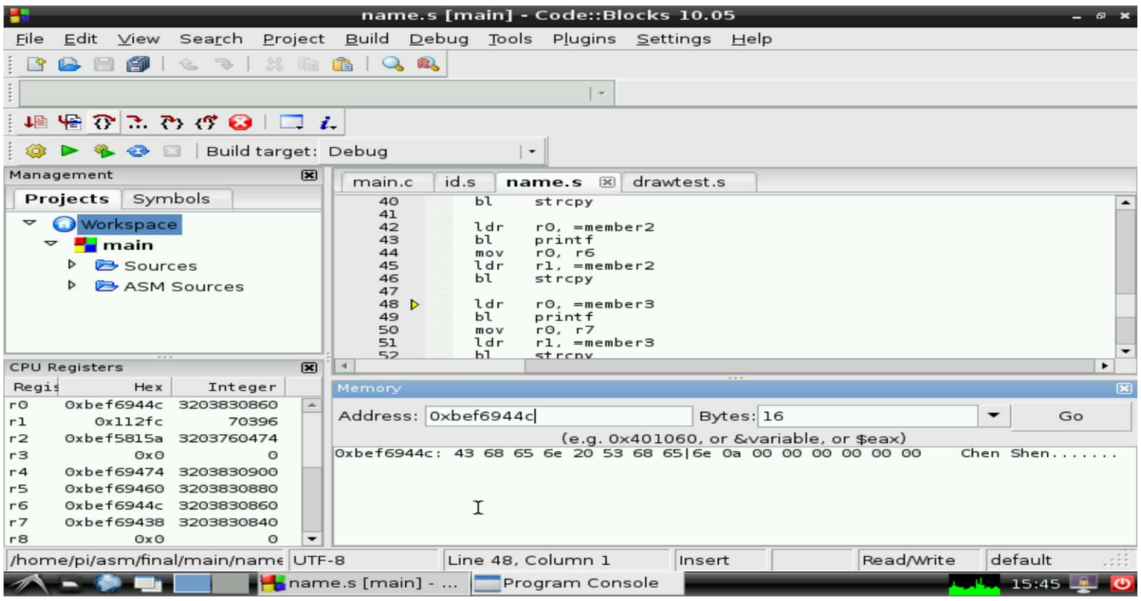
✦Project1 NAME 函數



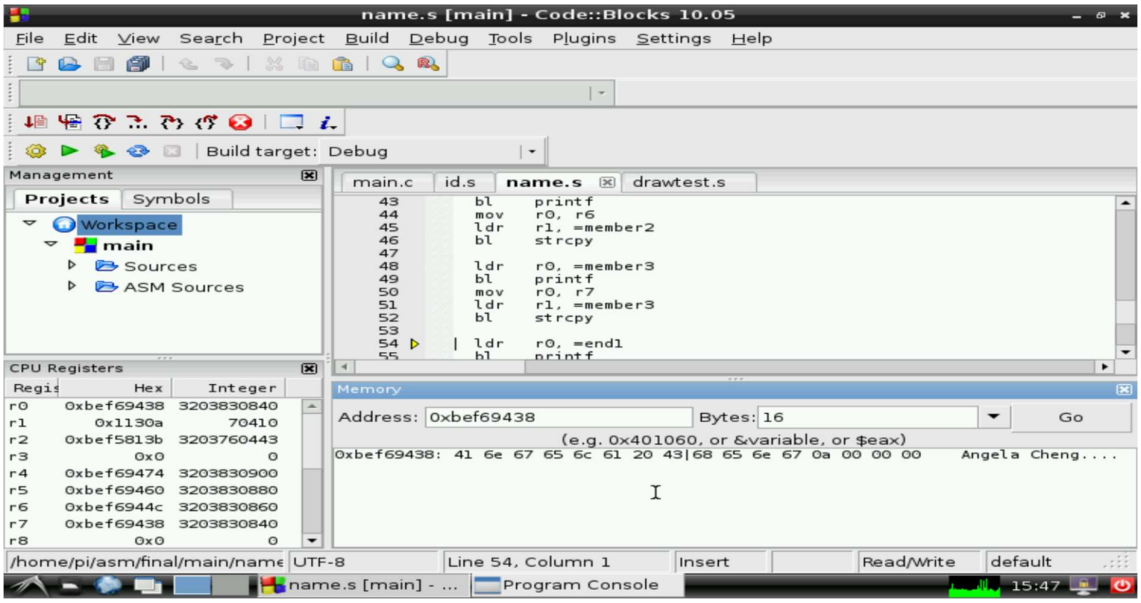
NAME Team Start：54 End：32



NAME Name1 Start：55 End：67

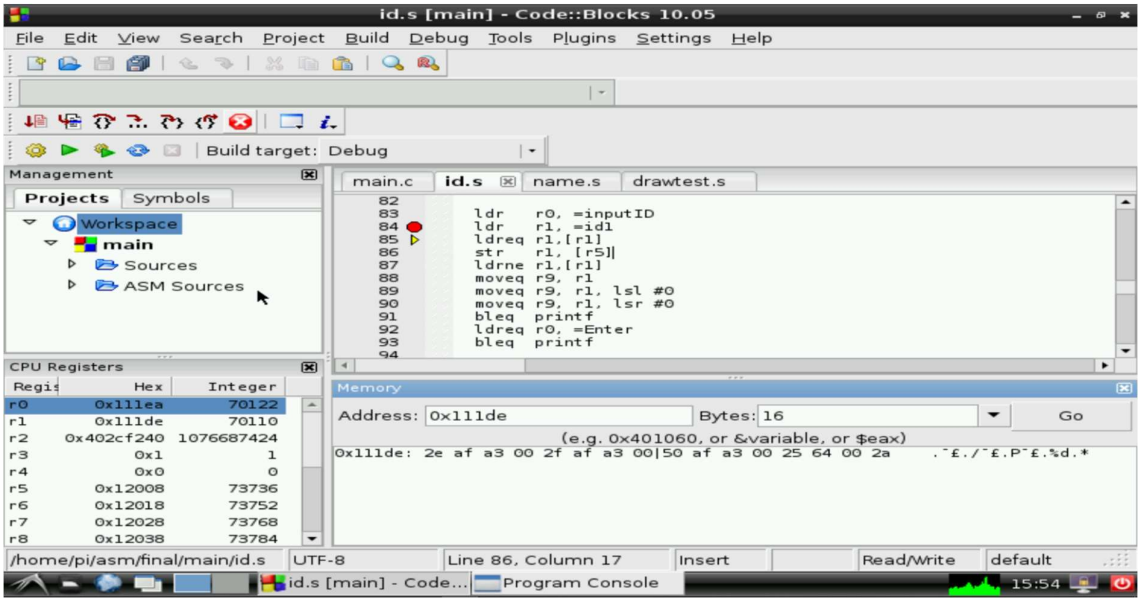


NAME Name2 Start：43 End：6e

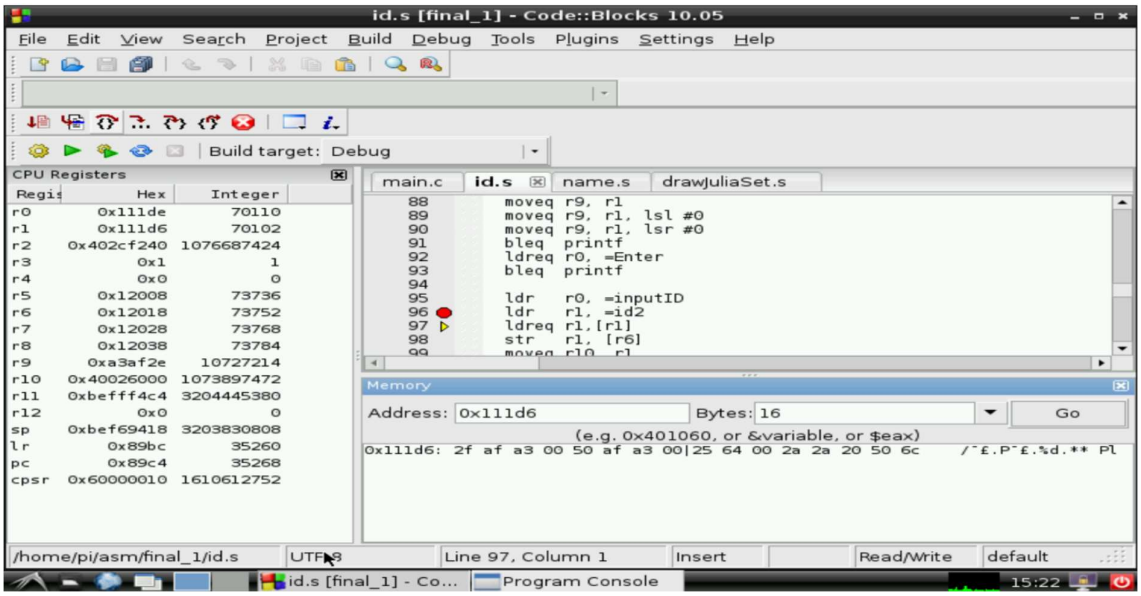


NAME Name3 Start：41 End：67

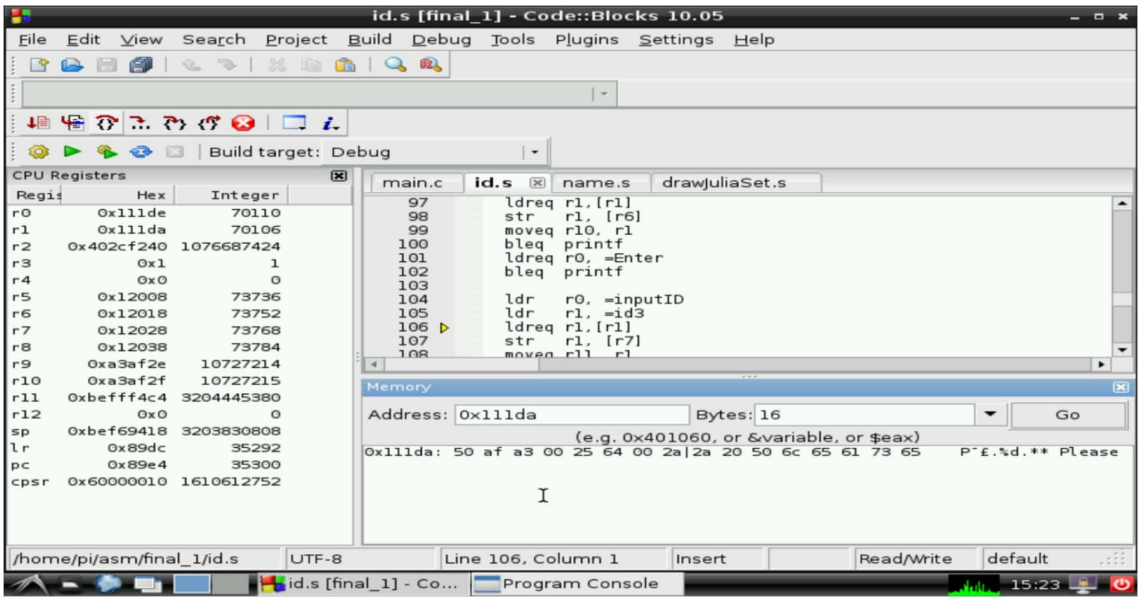
✦Project2 ID 函數



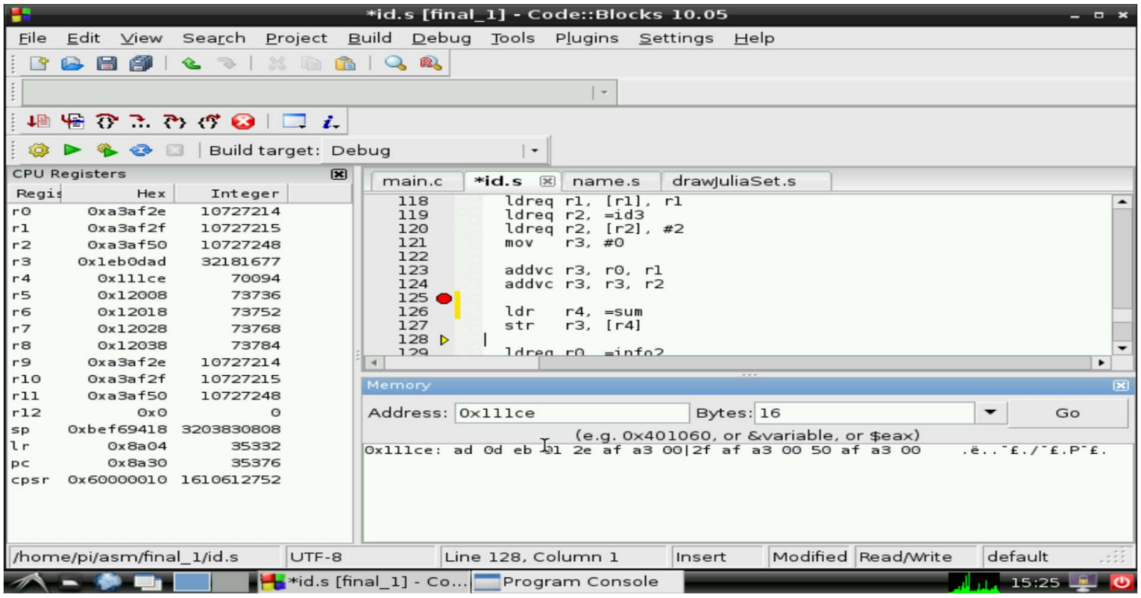
ID id1 Start：2e End：00



ID id2 Start：2f End：00

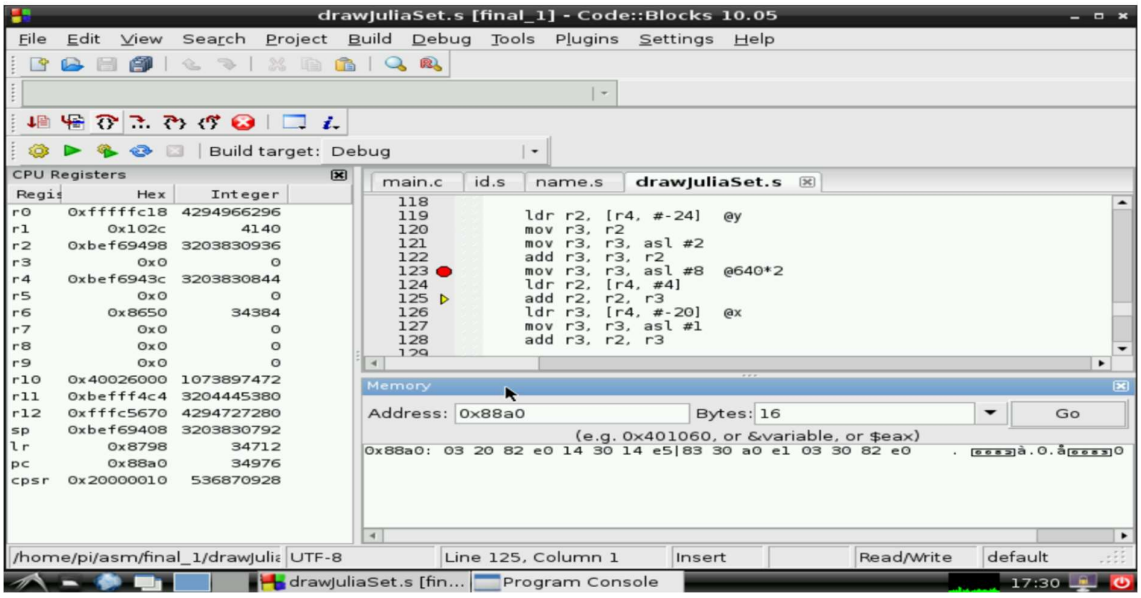


ID id3 Start：50 End：2a



ID sum Start：ad End：00

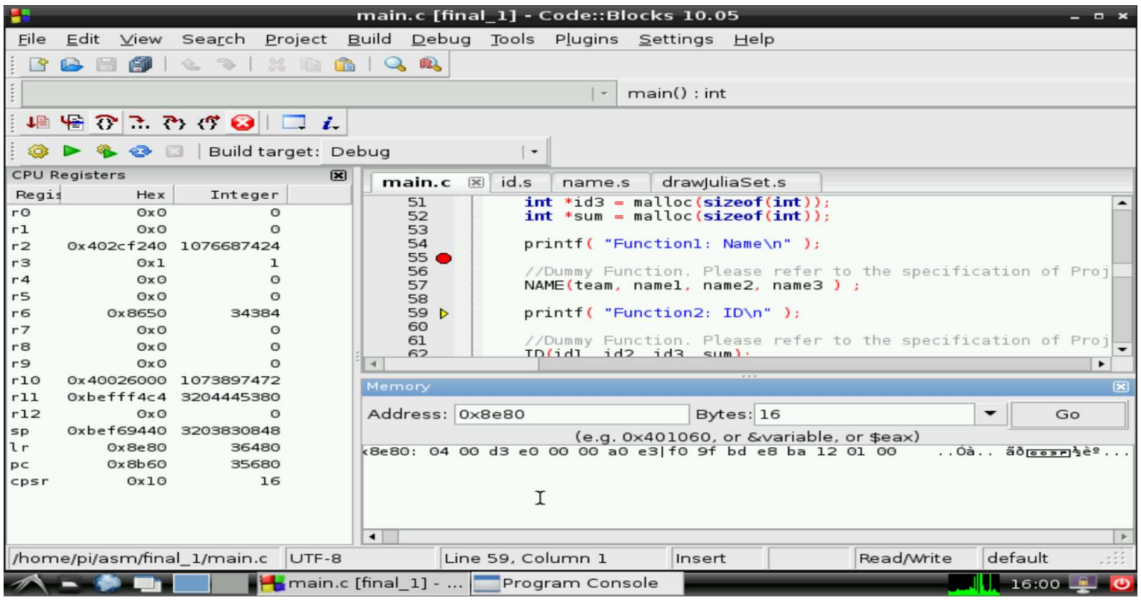
✦Project3 drawJuliaSet 函數



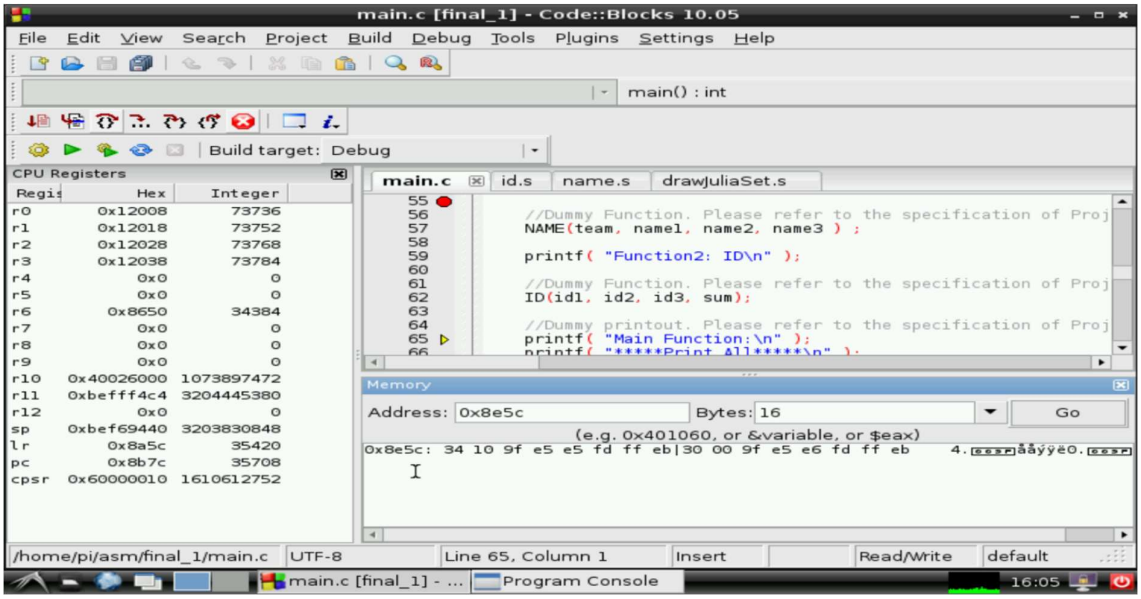
frame 的陣列起始位址: 03　結束位址: e5

圖中為陣列(0,0)區塊

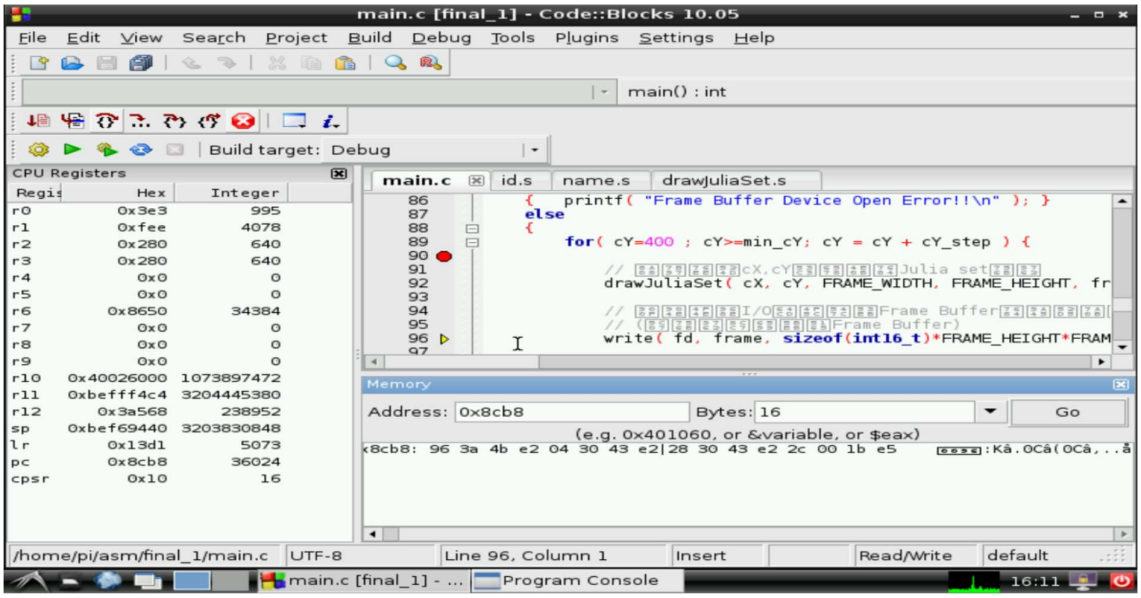
✦Project4 main 函數



main　NAME　Start：04　End：e3



main ID Start：34 End：eb



main drawJuliaSet Start：96 End：e2

四、討論&結論

（以下問題皆為本組撰寫程式時所遇到的事件）

Q:C與組合語言該如何回傳參數?

A:後來查詢資料發現暫存器依照r0到最後分別為 函數名(r0,r1,r2,r3)

Q:C、組合語言搭配，沒有復原pc造成Segmentation fault.

A:發現因為組合語言與C搭配的話，不復原pc，我們推測pc為控制整個程式的暫存器位置，若讓他繼續動，那這個函數通常會繼續跑，沒有結束的一天，除非出了甚麼意外碰巧跳出來了。

Q:研究JuliaSet的部分一開始寫了一些，圖完全出不來，是如何解決的?

A:翻了課本發現gcc反組譯，參考了反組譯的內容編寫。

Q:分工表

鄭珮慈

程式編寫

資料搜尋

書面報告

洪友祥

程式編寫

資料搜尋

書面報告

沈家丞

程式編寫

資料搜尋

書面報告

A:為何分工表一樣，因為組員每個人都有認真為這份報告作出貢獻，只是比例有可能有點差別。

五、未來展望

洪友祥

這次的final project真的花了我們好幾天，光在C呼叫組合語言就搞了一天了，然後將組語修改成C可以接受的內容，然後還有最難的JuliaSet,從完全不知道怎麼起手開始一點一點學習，雖然真的很難很累，但真的很有收穫的一門課程。

沈家丞

雖然不想欠過年，但無奈還是沒能在跨年前完成這次的期末project。從期中project之後有明顯感受到自己對組語的掌握有越來越好的趨勢，雖然JuliaSet又讓我知道還有很多進步的空間，而在與隊友們討論的同時，也理解了許多本來還不清楚的地方，希望期末考能因此考出理想的成績。

鄭珮慈

透過這次的組語期末project，讓我對組語的架構與指令運作有更深入的了解，而不是只有背書考試這種制式化的學習方式而已。而在撰寫程式的過程中，因為採取分組的方式進行，在遇到概念上無法釐清，甚至是編譯數次都無法通過時，也能和組員討論、交流想法，讓完成期末project的目標不再如此遙不可及。