

題目：期中 project

111 學年度第 1 學期

老師：朱守禮 老師

學生：徐梓薰 11027114

李若菱 11027133

游婕歆 11027149

程式說明

1. name.s :

我們先將 name 的字串，組別、三個組員的名字和一些標題，分別叫做 msg123...、No、member1、2、3，存入 data 段。並且在 text 段分別宣告全域的 name、No、member1、2、3。

name:

一開始我們先把 lr 備份到 stack 裡，因為在 bl printf 時會改變 lr 的位置。接著我們把 masl 的位置給 r0 後並 bl printf 印出裡面的內容，剩下的 data 段資料步驟同上，整個程式結束後 return 0，告訴系統我們成功執行程式，最後還原 lr 回到原本指令的下一行。

2. id.s :

先把所需的字串、存放要輸入的 inputid1、2、3，存入 data 段。並且在 text 段分別宣告全域的 id、inputid1、2、3、sum、msg3。

id: 一開始我們把 enter1(%d)的位置給 r0，再把 input_id1 的位置給 r1，然後 bl scanf 讀入使用者輸入的內容並把內容存入 inputid1，並在第五行加入老師指定的指令 subs r1, pc, lr(當不確定是正還是負數或知道是負數時使用 subs 若為負 cpsr 的 n(negative)會亮起)，並在他的下一行加入 adds r1, r1, lr 把它抵銷掉，之後利用和讀入 inputid1 一樣的方式讀入 inputid2，接著把 inputid1 的位址給 r1，去 memory(r1 裡的位置)裡把得到的值給 r1，再以同樣的方式處理 inputid2，並把它們相加，然後再把 input_id3 讀入，再把他們相加。讓使用者輸入 command，若是 p 就要跳到 runp:把三個 id 加起來的值印出來，若不是則結束。整個程式結束後 return 0，最後還原 lr 回到原本指令的下一行。

3. main.s :

先把所需的字串，存入 data 段。並且在 text 段分別宣告全域的 main。

main: 一開始我們先把 lr 備份到 stack 裡，因為在 call printf 時會改變 lr 的位置。一開始呼叫 name，再呼叫 id，按照順序印出組別、學號、姓名，最後在印出學號的總和結束。整個程式結束後 return 0，最後還原 lr 回到原本指令的下一行。

設計重點說明

1.

第五行加入老師指定的指令 subs r1 , pc, lr , 當不確定是正還是負數或知道是負數時使用 subs , 若為負 cpsr 中的 n(negative)會亮起 , 並在他的下一行加入 adds r1, r1, lr 把它抵銷掉。

2.

使用 bl printf 印標題、組別、組員姓名等和 bl scanf 讀入學號及 command

3. 定址模式

```
ldr r1, [r1] (register immediate)
ldr r1, [r1], #0 (immediate pre-indexed)
ldrne r8, [r8, #0] (immediate offset)
ldrne r8 [r8, r5] (register offset)
```

4.

```
addeq r4, r1, r2 (Z set (=))
addge r4, r4, #7 (Z set and V set, or N clear and V clear (>=))
ldrne r8 [r8, #0] (Z clear (!=))
```

5.

```
addeq r4, r1, r2 (r4 = r1 + r2)
mov r0, r1 (r0 = r1)
subs r1 , pc, lr (r1 = pc - lr )
```

程式驗證結果

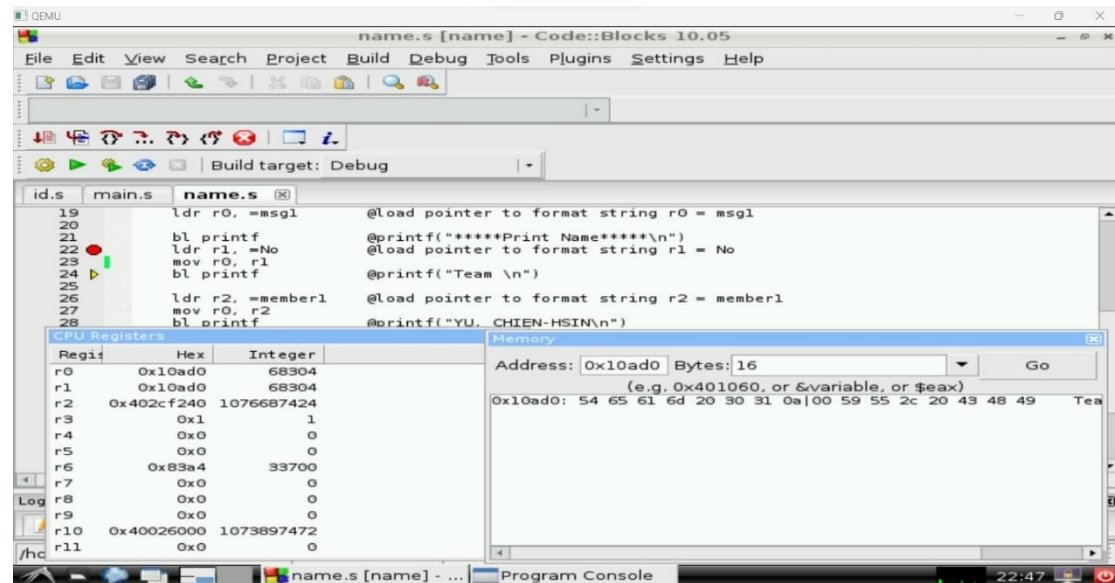
label	內容	起始位址	結束
No	Team 01	0x10ad0	0x10ad6
Member1	Yu, Chieh-hsin	0x10ad9	0x10ae6
Member2	Hsu, tzu-hsun	0x10ae9	0x10af5
Member3	Li, jo-ling	0x10af8	0x10b02
Inputid1	11027149	0x109d6	0x109d9
Inputid2	11027114	0x109da	0x109dd
Inputid3	11027133	0x109de	0x109e1
Sum	33054396	0x109e2	0x109e5
name		0x85dc	0x85e0(返回位址)
id		0x85e8	0x85ec(返回位址)

記憶體位置看法

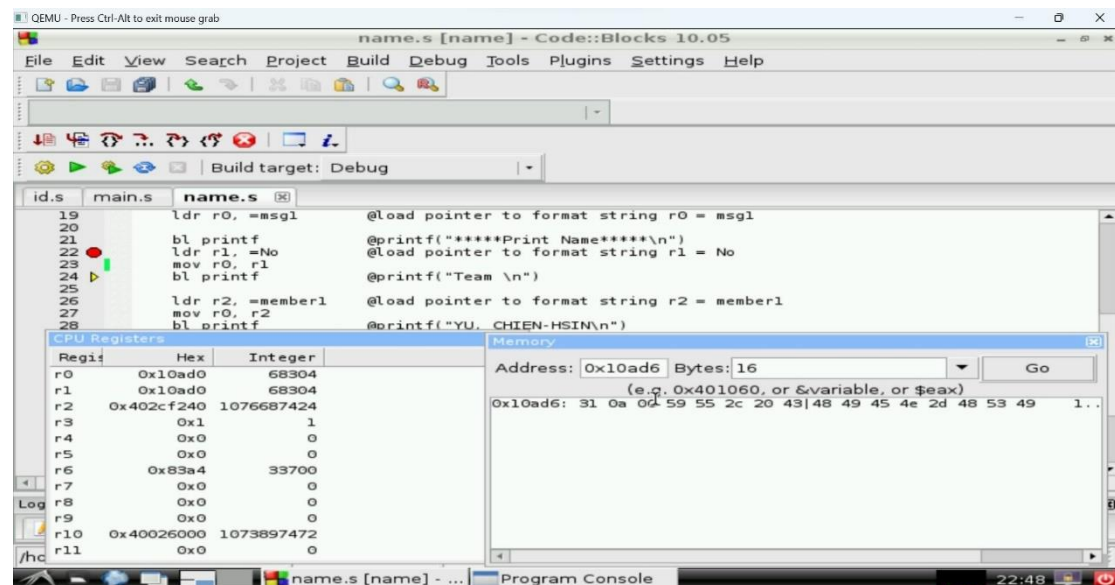
在看 inputid1, 2, 3 時，因為一個 int 是 4 個 bytes，在計算它的起始跟結束位置時是起始加上 3。

在看字串時(No、member1、2、3)，因為它是利用 char 來看，一個 char 是 1byte，在計算字串結束位置時，有幾個 char 就要往後加幾。

No
開始

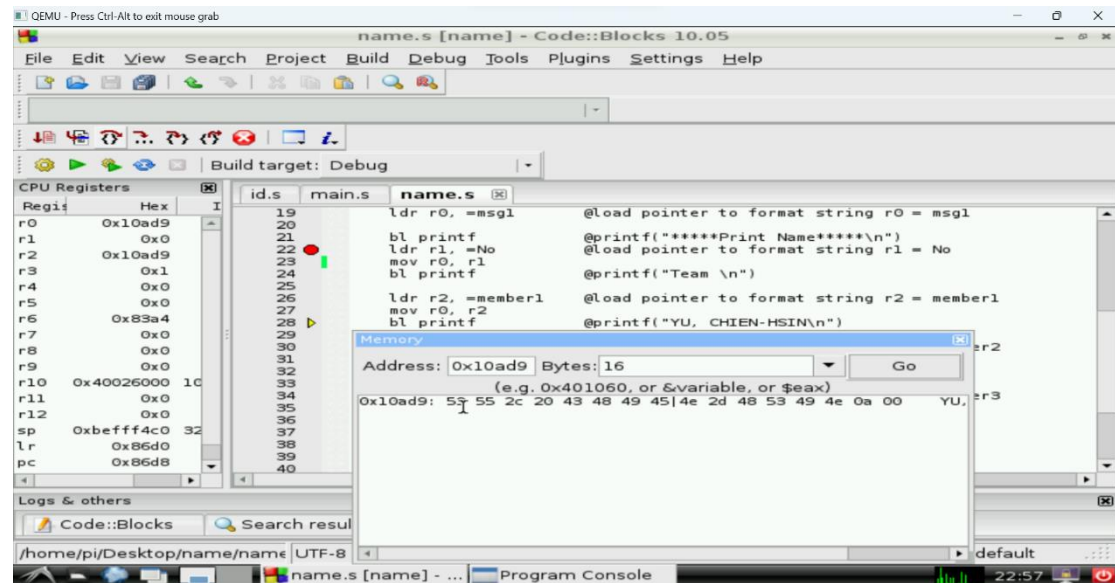


結束

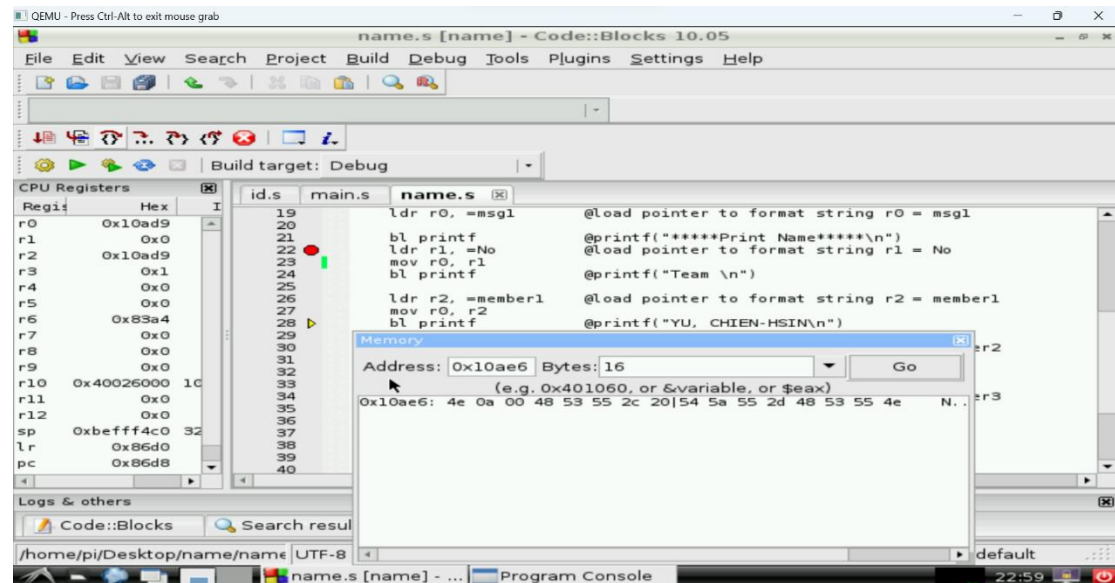


Member1

開始

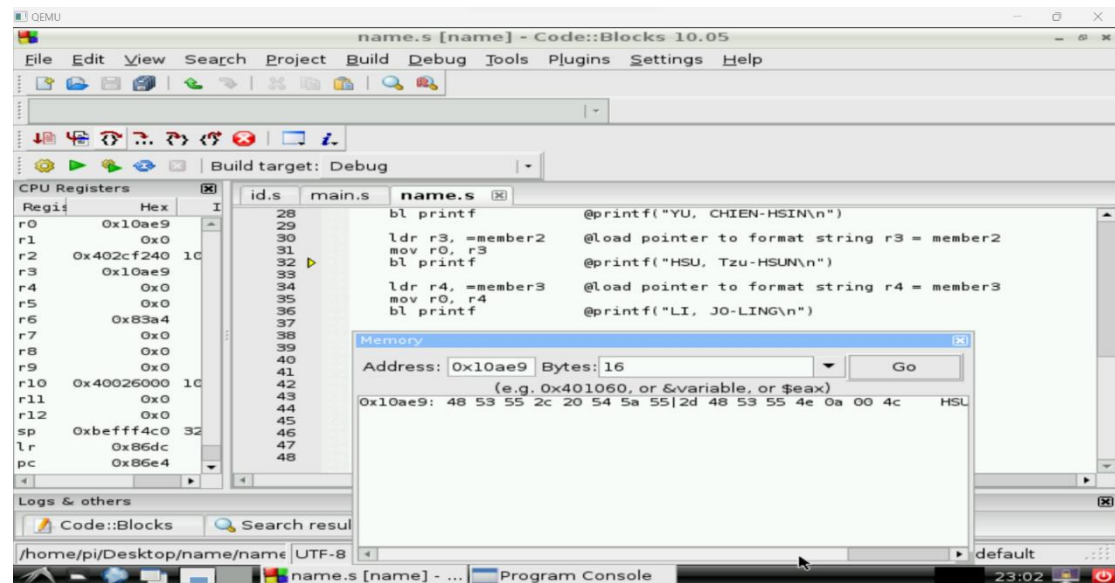


結束

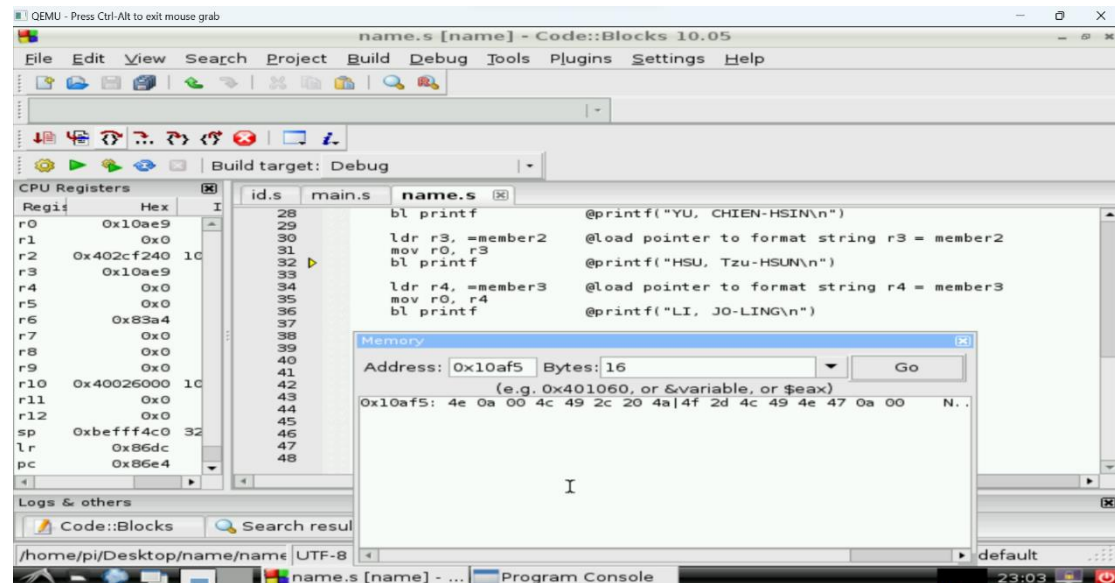


Member2

開始

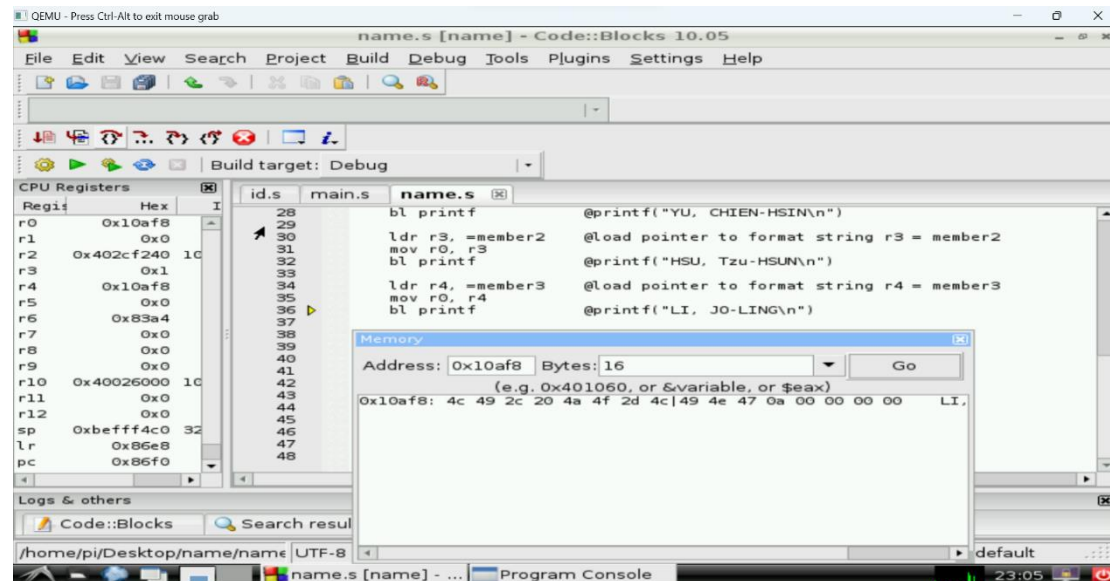


結束

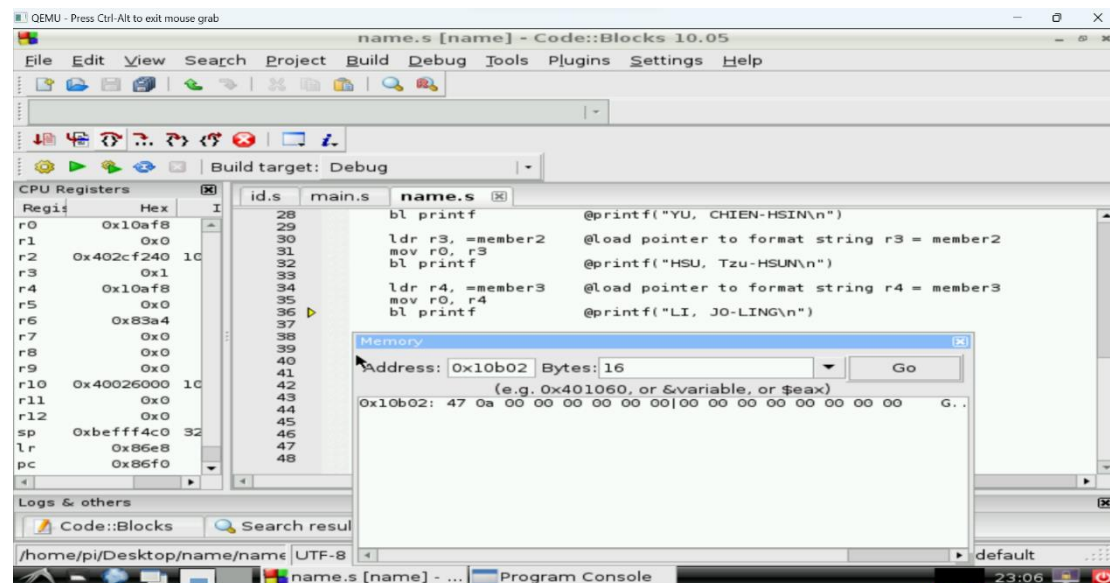


Member3

開始

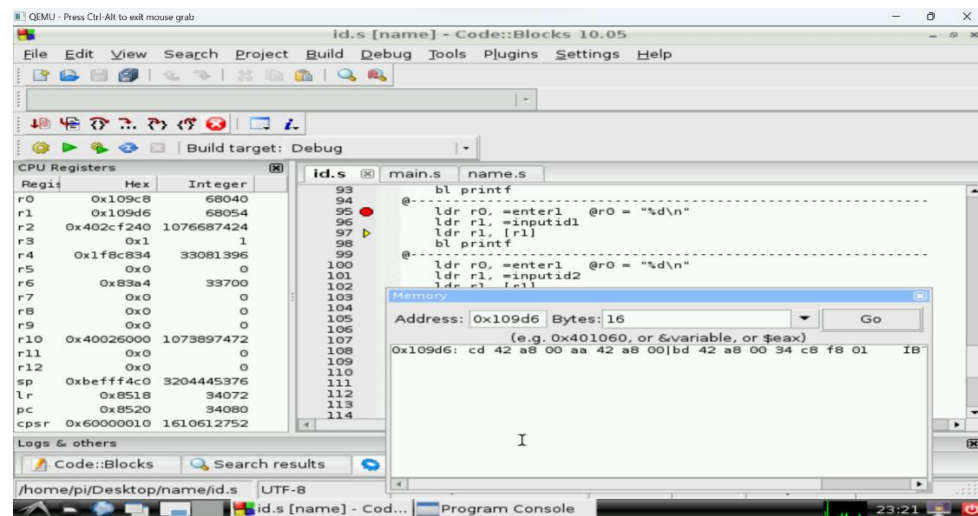


結束

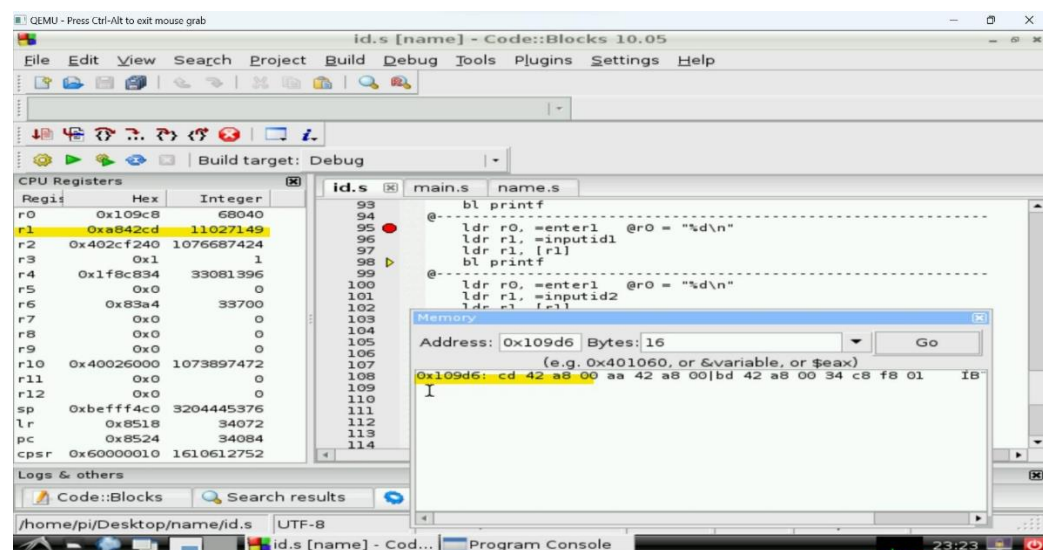
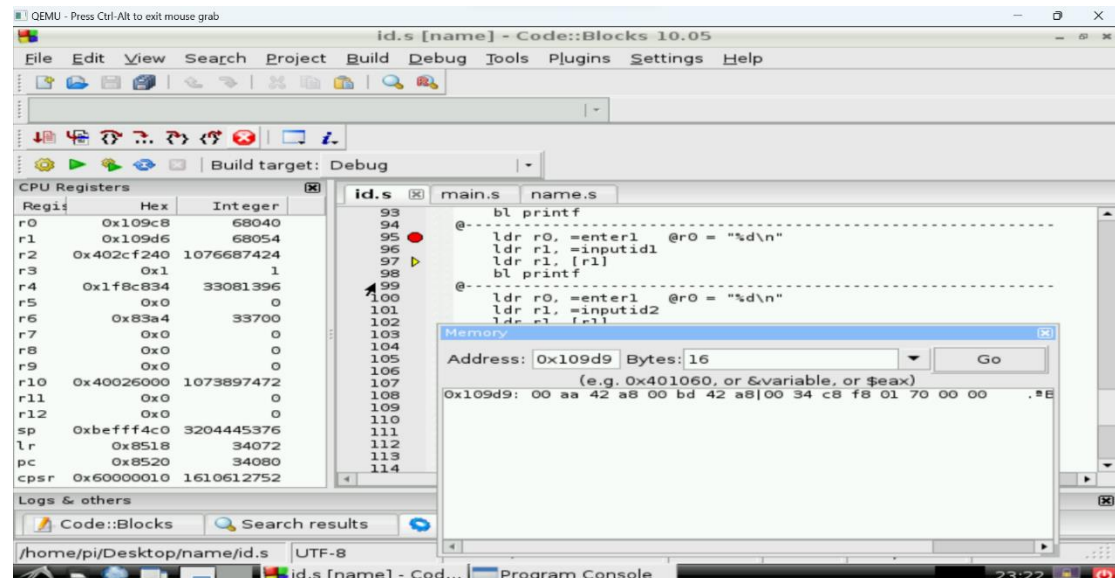


Inputid1

開始

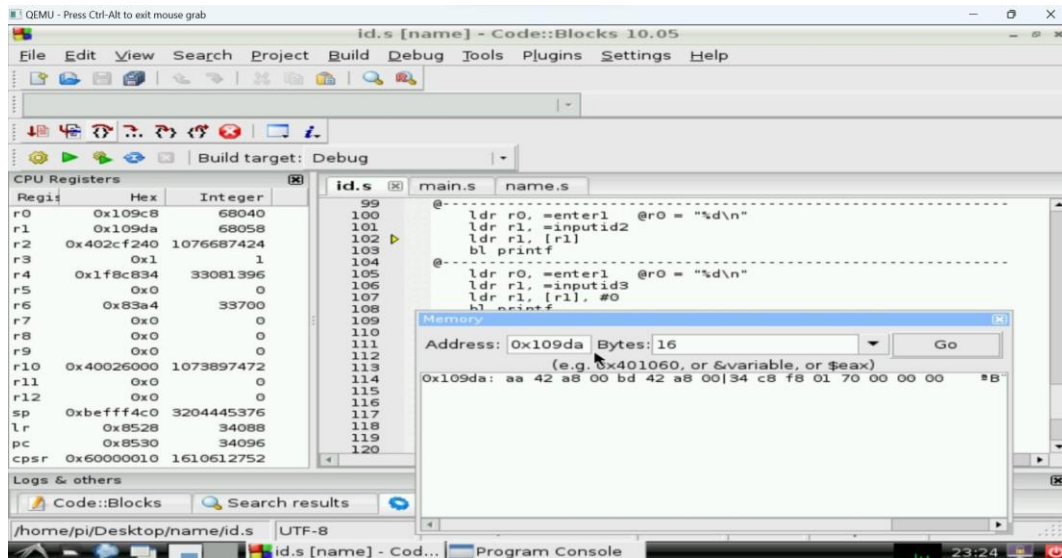


結束

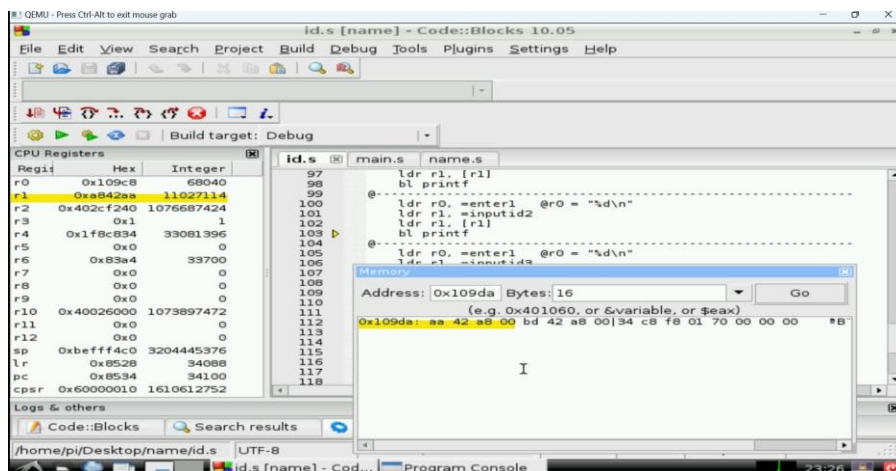
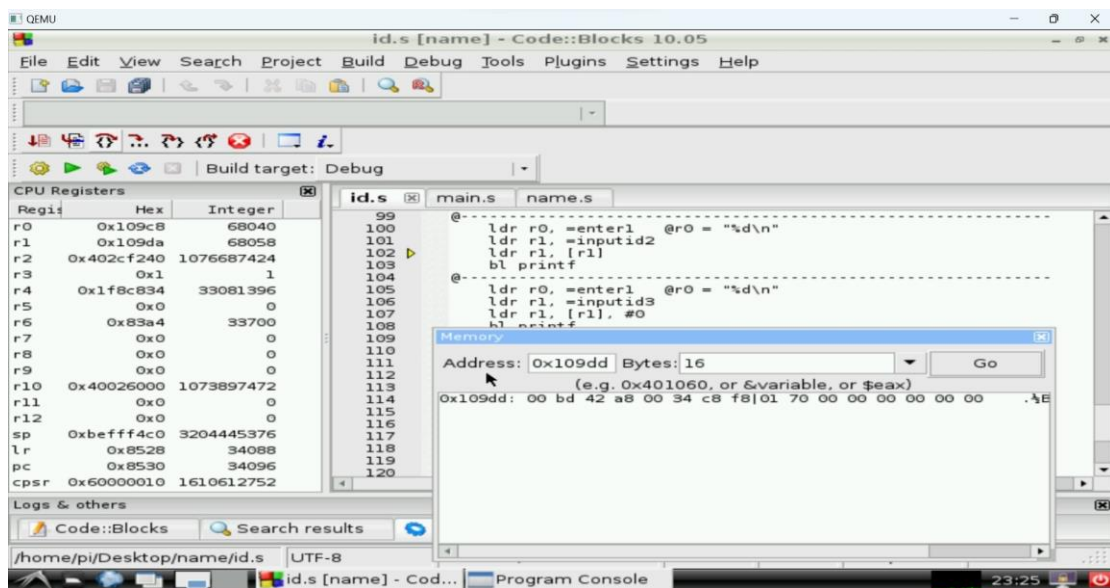


Inputid2

開始

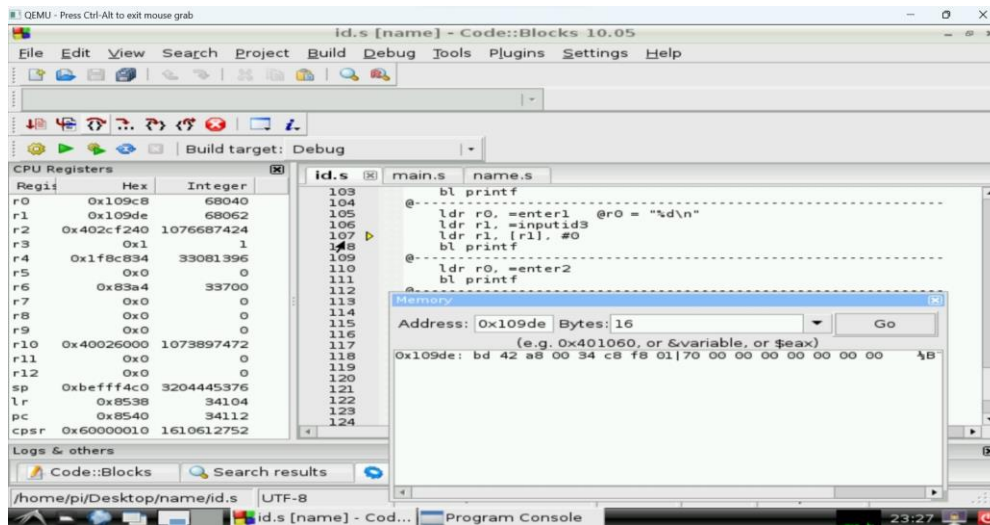


結束

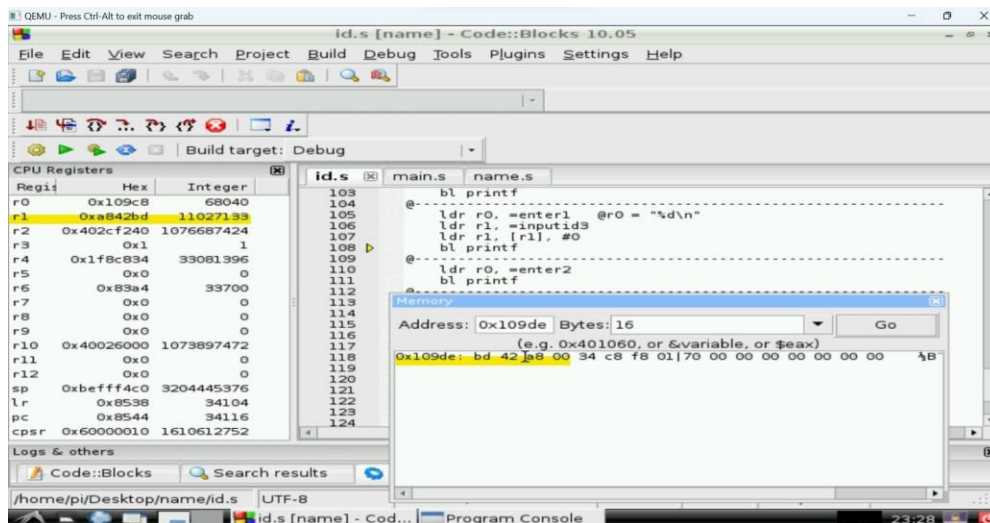
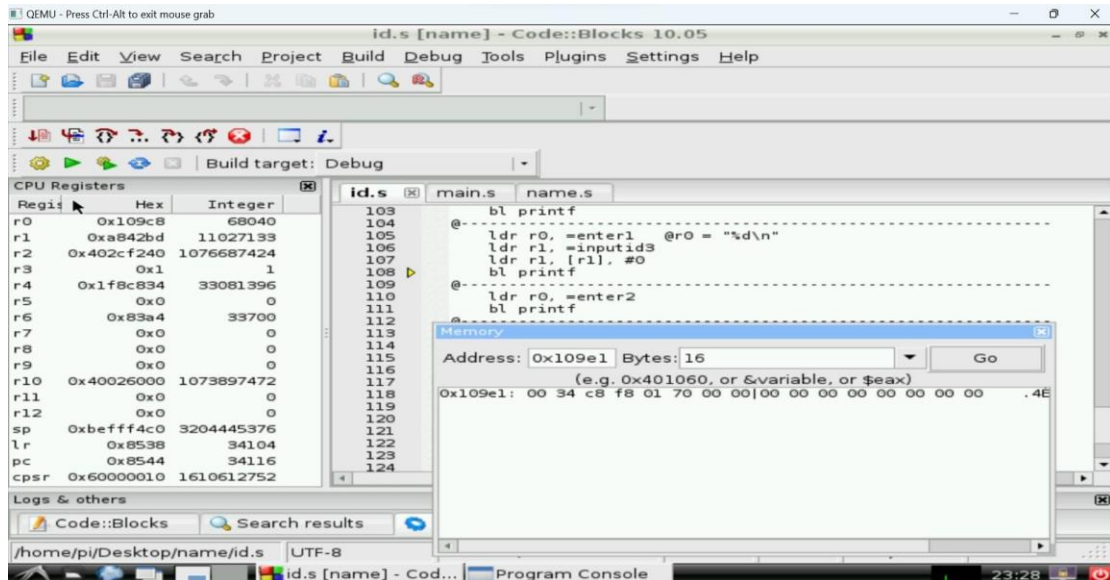


Inputid3

開始

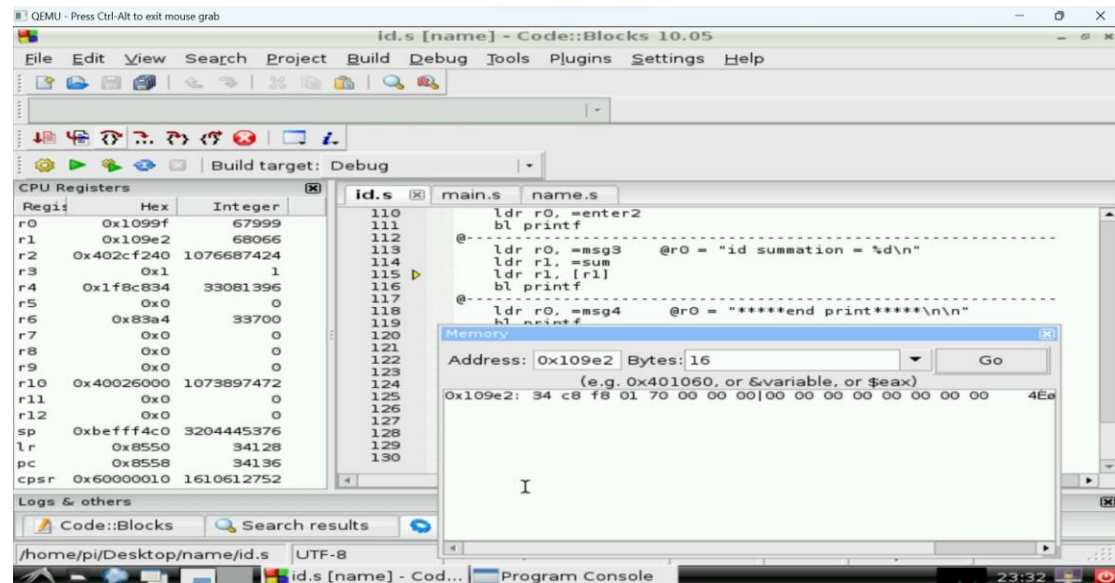


結束

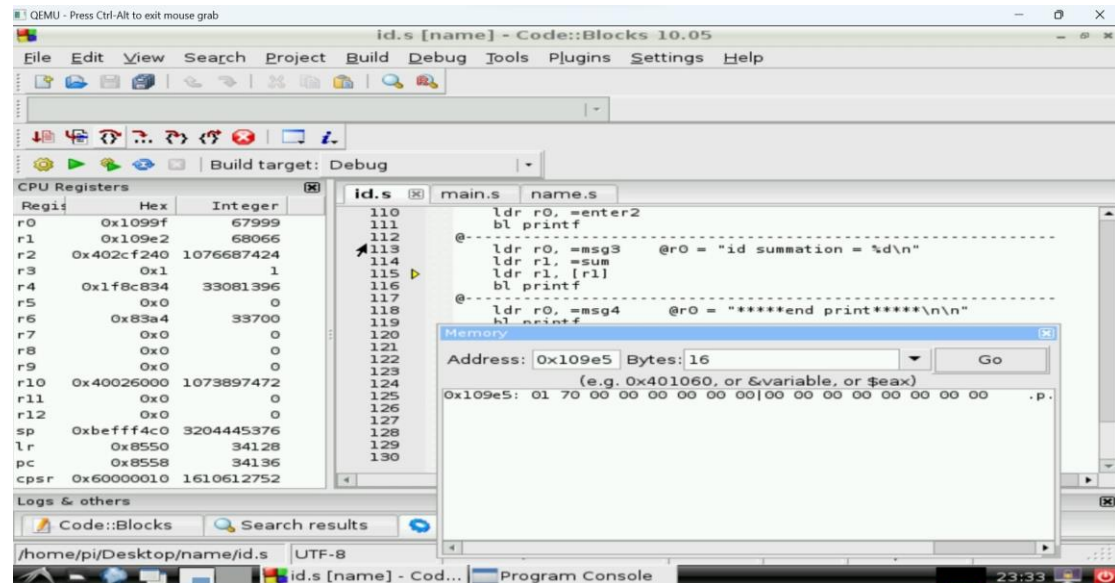


Sum 總和

開始

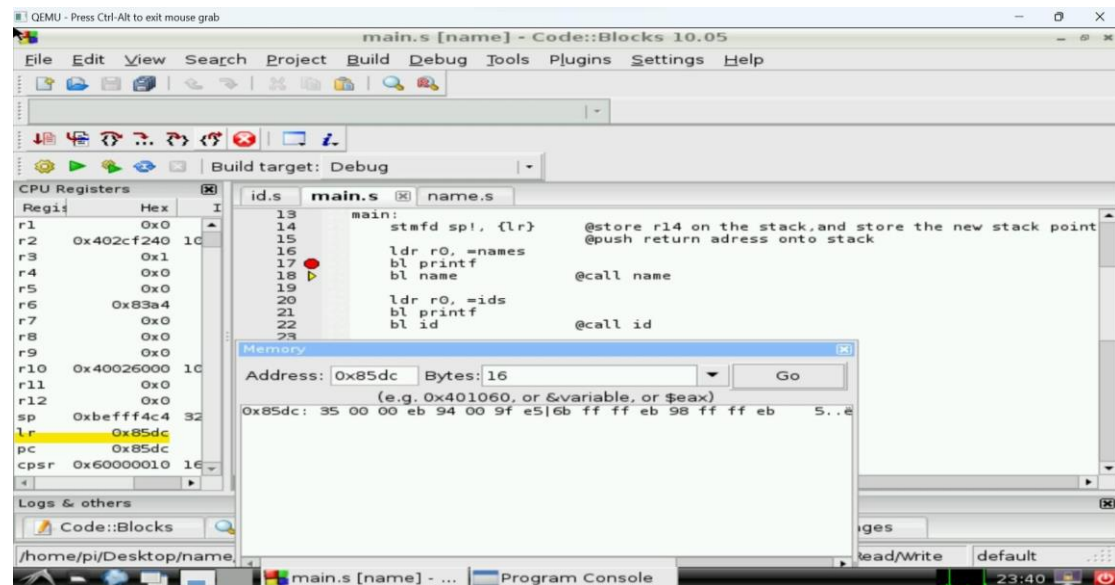


結束

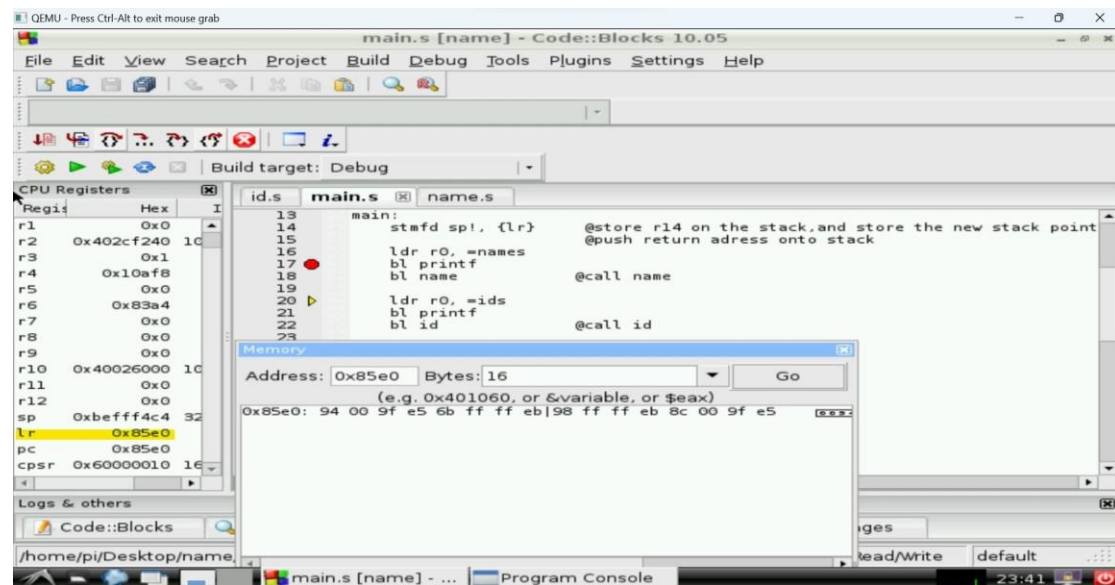


Name.s

所在位址

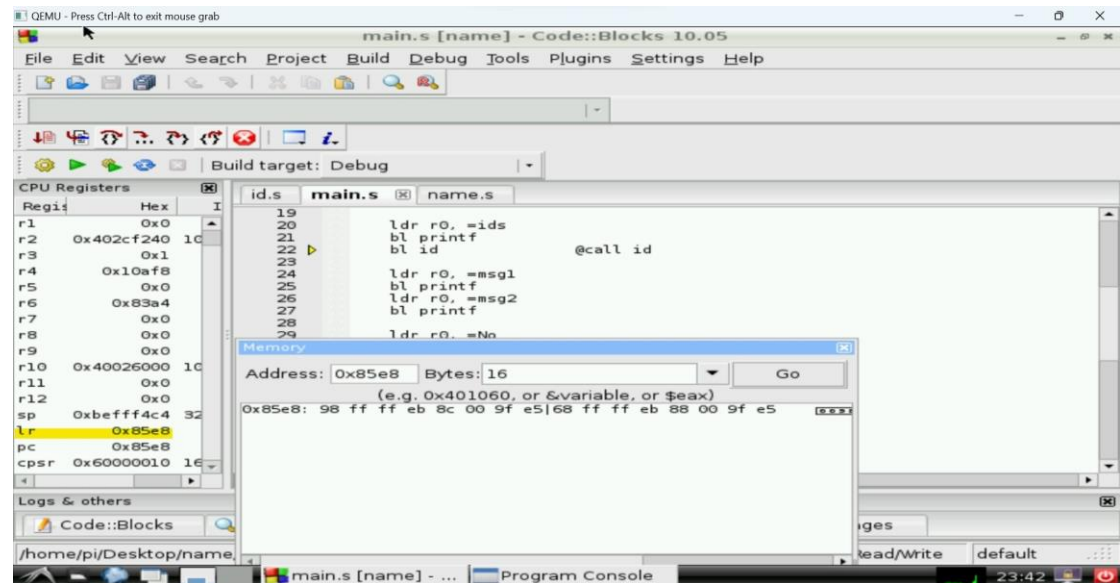


返回位址

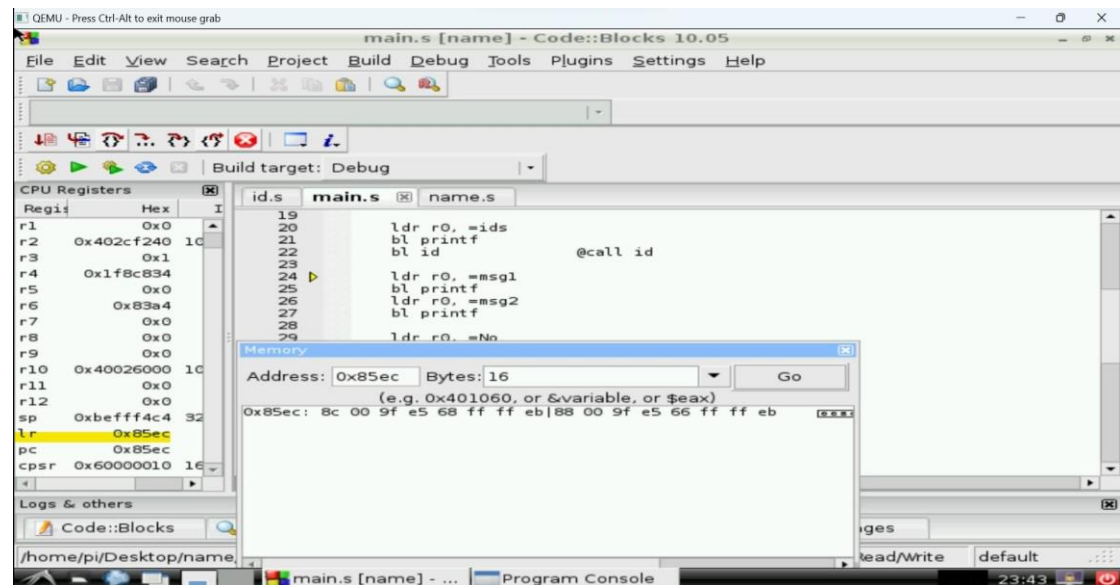


id

所在位址



返回位址



心得感想

一開始看完題目後完全不知道該如何下手，在寫 name.s 的時候發現應該跟課本 print “hello world” 應該是差不多的寫法於是就照著寫，就寫出來了但還是不太理解每一行到底是什麼意思，到寫了 id.s 的時候才比較理解為什麼要寫這些指令，就不是照抄而已，有慢慢在理解。在寫 main.s 時因為只是在呼叫前面的兩個函式所以簡單很多寫得比前兩題快。

分工

程式撰寫及書面報告都是三個組員一起討論完成。