

Report #13

Group 10

吳秉鴻(108502572)、楊佳峻(108502571)

1. 說明

- 使用函式 GetStdHandle 取得 consoleHandle 代號

```
17 ; Get the Console standard output handle:
18     INVOKE  GetStdHandle, STD_OUTPUT_HANDLE
19     mov     consoleHandle, eax
```

- 初始化: 將 xyPos 設為 xyInit 的值

```
21 ; 設定回到起始位置
22 ; mov xyInit to xyPos
23 INITIAL:
24     mov     ax, xyInit.x
25     mov     xyPos.x, ax
26
27     mov     ax, xyInit.y
28     mov     xyPos.y, ax
```

- 清空螢幕，再將游標指到 xyPos 的位置

```
30 v START:
31     call    ClrScr
32 v     INVOKE  SetConsoleCursorPosition,
33         consoleHandle,
34         xyPos
```

- 讀入一個字元，判斷上下左右(W, S, A, D)還有 Esc

```
36      call    ReadChar
37      .IF     ax == 1177h ;UP
38          sub  xyPos.y, 1
39      .ENDIF
40      .IF     ax == 1F73h ;DOWN
41          add  xyPos.y, 1
42      .ENDIF
43      .IF     ax == 1E61h ;LEFT
44          sub  xyPos.x, 1
45      .ENDIF
46      .IF     ax == 2064h ;RIGHT
47          add  xyPos.x, 1
48      .ENDIF
49      .IF     ax == 011Bh ;ESC
50          jmp  END_FUNC
51      .ENDIF
```

- 判斷邊界: x: [1, 79], y: [1, 24]，如果超出就減 1，如果少於就加 1

```
53      ; 檢查作完上下左右後有沒有超過限制邊界
54      .IF     xyPos.x == 0h ;x lowerbound
55          add  xyPos.x, 1 ; 要改成超過邊界停留在原位
56      .ENDIF
57
58      mov     ax, xyBound.x ; 註：比較不能用雙定址，故將其中一個轉成 register
59      .IF     xyPos.x == ax ;x upperbound
60          sub  xyPos.x, 1 ; 要改成超過邊界停留在原位
61      .ENDIF
62
63      .IF     xyPos.y == 0h ;y lowerbound
64          add  xyPos.y, 1 ; 要改成超過邊界停留在原位
65      .ENDIF
66
67      mov     ax, xyBound.y
68      .IF     xyPos.y == ax ;y upperbound
69          sub  xyPos.y, 1 ; 要改成超過邊界停留在原位
70      .ENDIF
```

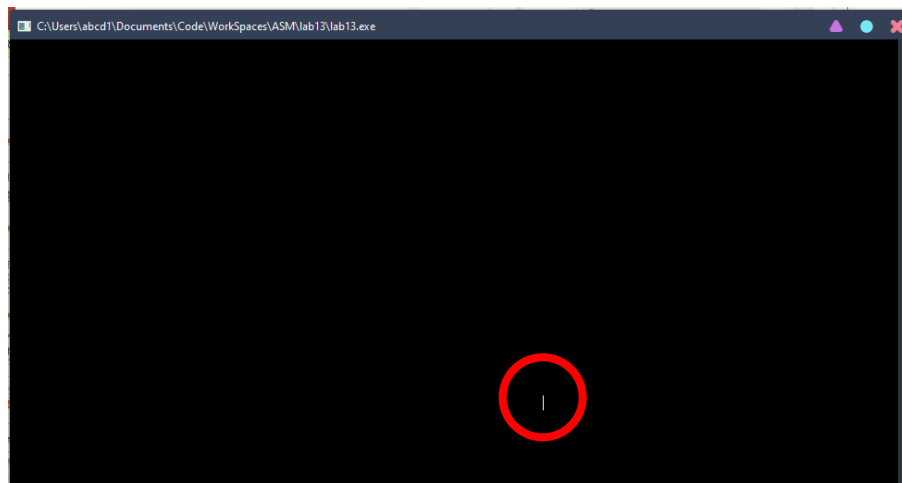
- 走完所有 .IF 則重新讀字元判斷。END_FUNC 來自輸入 Esc 後

結束 main PROC

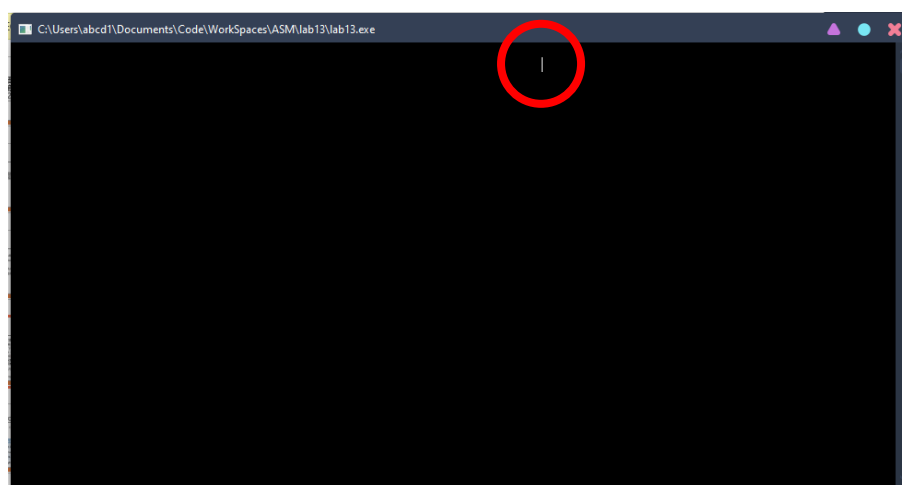
```
72      jmp      START
73
74  END_FUNC:
75      exit
```

2. 結果

- 初始(72, 24)



- W 按到底(72, 1)



3. 心得

這次作業相較前面有很大的成就感，挫折感比較少而且用到的東西都是有用過的。當然很開心這次是最後一次 LAB，之後就可以專心做專題了。好期待大家可以做出甚麼東西，感覺心中就充滿了動力。