報告標題: **lab13,** 組別: **48,** 姓名學號: 曾千芸**109504501** 賈子悅**110502514**

- **screenshots of the code**

```
INCLUDE Irvine32.inc
.data
consoleHandle    DWORD ?
xyInit COORD <2,15> ; 起始座標
xyBound COORD <80,25> ; 一個頁面最大的邊界
xyPos COORD <2,15> ; 現在的游標位置
main EQU start@0
.code
main PROC

; Get the Console standard output handle:
    INVOKE GetStdHandle, STD_OUTPUT_HANDLE
    mov consoleHandle,eax

; 設定回到起始位置
INITIAL:
    mov ax,xyInit.x
    mov xyPos.x,ax
    mov ax,xyInit.y
    mov xyPos.y,ax
START:
    call ClrScr
    INVOKE SetConsoleCursorPosition, consoleHandle, xyPos
    call ReadChar
    .IF ax == 1177h ;UP
        sub xyPos.y,1
    .ENDIF
    .IF ax == 1F73h ;DOWN
        add xyPos.y,1
    .ENDIF
    .IF ax == 1E61h ;LEFT
        sub xyPos.x,1
    .ENDIF
    .IF ax == 2064h ;RIGHT
        add xyPos.x,1
    .ENDIF
    .IF ax == 011Bh ;ESC
        jmp END_FUNC
    .ENDIF

    ; 檢查作完上下左右後有沒有超過限制邊界
    .IF xyPos.x == 0h ;x lowerbound
        add xyPos.x,1 ; 超過邊界停留在原位
    .ENDIF
    mov ax,xyBound.x ; 註：比較不能用雙定址，故將其中一個轉成 register
    .IF xyPos.x == ax ;x upperbound
        sub xyPos.x,1 ; 超過邊界停留在原位
    .ENDIF

    .IF xyPos.y == 0h ;y lowerbound
        add xyPos.y,1 ; 超過邊界停留在原位
    .ENDIF
    mov ax,xyBound.y
    .IF xyPos.y == ax ;y upperbound
        sub xyPos.y,1 ; 超過邊界停留在原位
    .ENDIF

    jmp START
END_FUNC:
    exit
main ENDP

END main
```

explaination:
We apply “readchar” to input ascii code of w, s, a and d into ax. Next, based on the conditions, move upward, downward, right and left. Then, we check if it surpasses the barrier or not, if it surpasses, it should go back one move.

The initial coordinate is (2 , 15) due to the (01mod79 +1 , 14mod24 +1).

- **thought of the lab:**

It is really challenging using the keyboard to move the coordinate of the dot, and it is the first time to change the output in the program. In today's lab practice, we learned how to move the keyboard. It helps us to apply these skills in our final project. Thanks to this experience, we learn a lot from it.