

Assembly Language Homework 3

學號：1102951

姓名：劉宗翰

1. 程式運行截圖：



```
E:\08_三下\02_組合語言\HW3>
This is a Morse Code Translator!
Please enter your string:
What's your name? 1<2*3.
Morse Code:
.-. .... - - .----. ... / -.- --- -.- .-. / -. .- -- . ----. / .----. @ ..--- @ ...-- .-.-
Would you like to proceed another translation (y/n)? |
```

- 設計概念：Morse Code 的陣列依 ASCII Code 的順序由小到大排列，並且由”Space”（ASCII Code = 20h）開始，至”_”（ASCII Code = 5h）結束，在陣列中，若有 Morse Code 無法表示的字元，則以”@”儲存於陣列中，若有特殊要求，如 Space，則以”/”儲存於陣列中，且陣列都以 DWORD 形式儲存，這是為了後續調用此陣列的方便性，但是，以此方式儲存，需要將 Morse Code 顛倒，如 A 原本的 Morse Code 為’.-.’，在陣列中就需以’.-.’的樣式儲存，最後，因為 Output Morse Code 字元間需要用 Space 做區隔，因此需要調用到 WriteChar，而 WriteChar 需調用到 AL 來印出指定字元，但題目中要求 Input String 的 OFFSET 需儲存於 EAX，這導致了 EAX 有兩個用途卻存在於同個 Procedure 中，因此，我在陣列中將所有元素都添加了” ”來取代 WriteChar 需調用到 AL 的問題。而作業中要求製作兩個 Procedure 分別為 lowerToCap 和 MorseTran，在 lowerToCap 中，判定 Input String 的個別字元是否大於 5Fh，若大於 5Fh，則減 20h 對應到大寫的 ASCII Code 或是錯誤”@”，這是因為 40h~5Fh 和 60h~7Fh 在 ASCII Code 轉為 Morse Code 中的陣列結構是一樣的，因此以大於 5Fh 減 20h 作為判定標準；而在 MorseTran 中，先將 lowerToCap 利用 ESI 已儲存的 Input String 起始位置恢復到 EAX，再進行運算，運算方式為，因為 Morse Code 陣列中以 ASCII Code = 20h 作為起始位置，所以先將 Input String 陣列中的數值減去 20h 以對應到 Morse Code 陣列的個別元素，又因陣列以 DWORD 形式儲存，每個 Morse Code 間相隔三個 DWORD，所以需乘上 12，最後再加上 Morse Code 陣列的起始位址來對應到該字元在 Morse Code 陣列中的正確位址，並且印出，再進行迴圈，最後跳出迴圈，並回到 Main Procedure 詢問是否再次輸入。

3. 程式碼：

```
1. INCLUDE Irvine32.inc
2.
3. BUFMAX = 128
4.
5. .data
6. introText      BYTE "This is a Morse Code Translator!", 0
7. enterText      BYTE "Please enter your string:", 0
8. outputText     BYTE "Morse Code:", 0
9. askStr         BYTE "Would you like to proceed another translation
(y/n)? ", 0
10. inputStr      BYTE BUFMAX+1 DUP(0)
11. morseCode      DWORD  ' /', null, 0, ' @', null, 0, '...-', ' .-',
    0, ' @', null, 0, '----', ' -..', 0, ' @', null, 0, ' @', null, 0, '---
    -', ' .-', 0, '----', ' -.', 0, '----', ' -.', 0, ' @', null, 0, ' @',
    null, 0, '....', ' --', 0
    ; SP ~ ,
12.              DWORD  '....', ' -.', 0, '----', ' -.', 0, '----',
    ' .', 0, '----', ' -', 0, '----', ' -', 0, '----', ' -', 0, '----', ' -
    ', 0, '....', ' -', 0, '....', ' .', 0, '....-', ' .', 0, '....-', ' .',
    0, '----', ' .', 0, '----', ' .', 0
    ; - ~ 9
13.              DWORD  '----', ' ..', 0, '----', ' .-', 0, ' @', null,
    0, ' @', null, 0, ' @', null, 0, '---.', ' ..', 0, ' @', null, 0
    ; : ~ @
14.              DWORD  ' .-', null, 0, '....', ' ', 0, '----', ' ',
    0, ' .-', null, 0, ' .', null, 0, '---.', ' ', 0, ' .-', null, 0,
    '....', ' ', 0, ' ..', null, 0, '----', ' ', 0, ' -.-', null, 0, '....',
    ' ', 0, ' --', null, 0
    ; A ~ M
15.              DWORD  ' .-', null, 0, ' ---', null, 0, '---.', ' ',
    0, '---.', ' ', 0, ' .-', null, 0, ' ...', null, 0, ' -', null, 0, '
    -..', null, 0, '----', ' ', 0, ' ---', null, 0, '---.', ' ', 0, '---.',
    ' ', 0, '....', ' ', 0
    ; N ~ Z
16.              DWORD  ' @', null, 0, ' @', null, 0, ' @', null, 0,
    ' @', null, 0, ' @', null, 0
    ; [ ~ _
17.
18. .code
```

```

19. main PROC
20.     mov EDX,OFFSET introText ; print "This is a Morse Code Translator!"
21.     call WriteString
22. keepGoing:
23.     call CrLf
24.     mov EDX,OFFSET enterText ; print "Please enter your string:"
25.     call WriteString
26.     call CrLf
27.     mov EDX,OFFSET inputStr ; input string
28.     mov ECX,BUFMAX
29.     call ReadString
30.     mov ECX,EAX ; store string size
31.     call lowerToCap ; change lowercase to uppercase
32.     mov EDX,OFFSET outputText ; print "Morse Code:"
33.     call WriteString
34.     call CrLf
35.     call MorseTran ; change string to morse code
36.     call CrLf
37.     mov EDX,OFFSET askStr ; print "Would you like to proceed another
translation (y/n)? "
38.     call WriteString
39.     call ReadChar
40.     cmp AL,79h
41.     je keepGoing
42.     exit
43. main ENDP
44.
45. lowerToCap PROC
46.     push ECX
47.     mov EAX,OFFSET inputStr
48.     mov ESI,EAX ; store the starting position of
input string to ESI
49.
50. lowercaseCheck:
51.     mov EBX,0 ; clean up EBX
52.     mov BL,[EAX]
53.     cmp BL,60h
54.     jb noChange

```

```

55.      sub BL,20h                ; lowercase change to uppercase
56. noChange:
57.      mov [EAX],BL
58.      inc EAX
59.      loop lowercaseCheck
60.
61.      pop ECX
62.      ret
63. lowerToCap ENDP
64.
65. MorseTran PROC
66.      push ECX
67.      mov EAX,ESI                ; restore the starting position of
        input string to EAX
68.
69. TransferToMorse:
70.      mov EBX,0
71.      mov BL,[EAX]
72.      sub BL,20h
73.      imul BX,12
74.      mov EDX,OFFSET morseCode
75.      mov ESI,EDX
76.      add ESI,EBX
77.      mov EDX,ESI
78.      call WriteString
79.      inc EAX
80.      loop TransferToMorse
81.
82.      pop ECX
83.      ret
84. MorseTran ENDP
85. END main

```