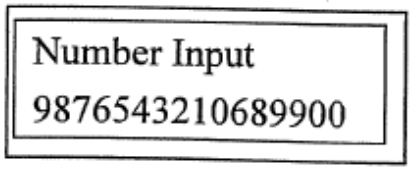
4-3 數字輸入(Number Input)

題目：

藉由 4X4 點矩陣鍵盤的操作，進行 Number Input 的數字輸入(輸入一串數字)。程式在 LCM 顯示功能選單畫面時，按下數字 ”3” 鍵進行數字輸入設定(Number Input)的功能，當按下數字鍵 ”3” 後，LCM 第一列的畫面立即出現 ”Number Input” 等文字內容，而 LCM 第二列則爲空白的狀態，接著等待操作者輸入 0~9 阿拉伯數字，一旦輸入一個阿拉伯數字後，這個數字就在 LCM 的第二列上立即顯示出來，而游標就會向右邊移動一位數，在此狀態下若繼續輸入 0~9 阿拉伯數字，那麼新的數字就繼續在 LCM 的第二列上往右邊排列顯示出來，最多顯示 16 個阿拉伯數字，若輸入超過 16 個數字之後，則屬無效數字不需要顯示出來；假若在數字輸入的功能狀態下按”Mu” 鍵，LCM 會立即返回功能選單的畫面。



數字輸入這題的題目較單純，所以先以這題做練習，參考程式位於 Number Input 資料匣中的 ni.a，程式內容入下：

```
***show character "3V" on LCD***
show_dt:
          mov    a,#33h          ;3 =>ascii code 33h
          mov    abuf,a
          call   wr_lcddat
          mov    abuf,#56h       ;V =>ascii code 56h
          call   wr_lcddat
          end
```

以上副程式的功能，在介紹如何將”3V”這 2 個字顯示於 LCD，真正在運用時，有比較實用的寫法。由表 4-3-1 可知，數字 0~9 和 LCD 要顯示的”0”~”9”字型剛好差 30H，所以當我們讀到鍵盤上的數字時，如果要將它轉成字型(ASCII 碼)顯示於 LCD，要事先加上 30H。這部分在等一下要將按到數字鍵的值顯示於 LCD 時會用到，記得要來這裡參考。

至於要顯示字母”V”，並不需要事先查其字型碼 56H，只要寫上”V”，組譯時組譯器會自動做轉換，這部分在類比數位轉換(ADC)那一題會用到。

表 4-3-1 LCD 字型對應表

30H

Upper 4 Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	ß	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	q	π
xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	-¹	y
xxxx1010	(3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)		-	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	゜	ö	

Note: The user can specify any pattern for character-generator RAM.

數字輸入程式設計說明：

步驟一：首先去尋找顯示數字輸入(Number Input)的那段副程式，這段程式目前的功能只有顯示”Number Input”，判斷按到”Mu”鍵時回到主畫面

```
NI_menu:
          mov     dptr,#NI_scr1
          call    sho_lcm1
          mov     dptr,#NI_scr2
          call    sho_lcm2
NI_scan:  call    scan_key
          mov     a,keynum
          cjne    a,#0ah,NI_scan      ;判斷是否為”Mu”鍵
          ret
```

步驟二：判斷是否為數字鍵 0~9，如果是，將其顯示於 LCD

```
NI_menu:
          mov     dptr,#NI_scr1
          call    sho_lcm1
          mov     dptr,#NI_scr2
          call    sho_lcm2
NI_scan:  call    scan_key
          mov     a,keynum      ;取回按鍵值
          clr     c
          subb    a,#10         ;將按鍵值減 10
          jnc     nn            ;判斷是否有借位(小於 10)，
                                  如果大於 10 就跳至 nn 判斷是否是"Menu"鍵
          mov     a,keynum      ;取回按鍵值
          add     a,#30h        ;加上 30H，調整成 ASCII 碼
          mov     abuf,a
          call    wr_lcddat     ;將數字字型顯示於 LCD
          jmp     NI_scan
nn:
          mov     a,keynum
          cjne    a,#0ah,NI_scan      ;判斷是否為”Mu”鍵
          ret
```

以上程式中，mov a,keynum 將按鍵值存入累加器 ACC 這行一共出現 3 次，因為經過減 10 或調整成 ASCII 碼，都會改變原來累加器 ACC 內的數值，所以必須重做一次才可以，不然

會出錯。

到此數字輸入(Number Input)功能已經完成，題意雖然說輸入超過 16 個字不需要顯示，因為 LCD 實際一行也只能顯示 16 個字，超過部分也看不到，所以那部分不需要寫程式去判斷。

以上透過減法(減 10)來判斷是否是數字鍵 0~9 的寫法，在號碼設定(Number Set)以及專家級認證的串列記憶體(MEM)這 2 題都會用到，千萬要記熟。