



## **North South University**

Department of Electrical Computer Engineering

### **Lab Assignment**

**Course name:** Microprocessor Interfacing and Embedded System

**Course code:** CSE331L

**Term:** Fall'19

**Instructor:** Dr. Sakhawat Hussain (SkH1)

**Lab Instructor:** M. A. Muhiminul Islam

**Date:** 23/12/19

Submitted by

**Md. Masudur Rahman**

**ID:** 1631189042

```
1  include "emu8086.inc"
2  .model small
3  .stack 100h
4
5
6  .data
7
8
9  .code
10
11
12  main proc
13
14      mov ax,@data
15      mov ds,ax
16
17      ;initialization
18      mov ax,0
19
20      ;input al = AAh, output will be same
21
22      mov al,0AAh
23
24      mov bl,10h
25      div bl
26
27      cmp ah,al ;for comparing lower and higher nibble
28      jz same
29      jmp notSame
30
31
32      same:
33          CALL PTHIS
34          db 13, 10, 'SAME',0
35          jmp end
36
37
38      notSame:
39          CALL PTHIS
40          db 13, 10, 'NOT SAME',0
41
42
43      end:
44          DEFINE_PTHIS
45          hlt
46
47
48
49
50
51
52
53  endp main
54  end main
```

```

1  include "emu8086.inc"
2  .model small
3  .stack 100h
4  .data
5      ;initializing 3*3 matrix
6      matrix db 1,2,3 ; row sum = 6 ;col sum = 12
7              db 4,5,6 ; row sum = 15 ;col sum = 15
8              db 7,8,9 ; row sum = 24 ;col sum = 18
9
10 .code
11 main proc
12     mov ax,@data
13     mov ds,ax
14
15     ;initialization
16
17     mov ax,0
18     mov si,0
19     mov di,3
20
21     row:
22         mov bl,matrix[si]
23         add al,bl
24         inc si
25         cmp si,10
26         jz mid
27         cmp si,di
28         jz result
29         jmp row
30
31     result:
32         CALL PTHIS
33         db 13, 10, 'Row Sum : ', 0
34         CALL PRINT_NUM
35         mov ax,0
36         add di,3
37         jmp row
38
39     ;column sum start from here
40     mid:
41         mov ax,0
42         mov si,0
43         mov di,6
44         mov cx,0
45
46     col:
47         mov bl,matrix[si]
48         add al,bl
49         cmp si,8
50         ja end
51         cmp si,di
52         jz colResult
53         add si,3
54         jmp col
55
56     colResult:
57         CALL PTHIS
58         db 13, 10, 'Column Sum : ', 0
59         CALL PRINT_NUM
60         mov ax,0
61         inc di
62         inc cx
63         mov si,cx
64         jmp col
65
66     DEFINE_PTHIS
67     DEFINE_PRINT_NUM
68     DEFINE_PRINT_NUM_UN
69
70     end:
71         hlt
72
73 endp main
end main

```

```

1  include 'emu8086.inc'
2  .model small
3  .stack 100h
4
5  .data
6      input db "ab123 cd$" ; alphabet = 4, digit = 3
7                          ; will not count space
8  .code
9
10 main proc
11
12     mov ax,@data
13     mov ds,ax
14
15     mov si,0
16     mov cx,0
17     mov dx,0
18
19     l1:
20         mov al,input[si]
21
22         cmp al,'$'
23         jz end
24
25         inc si
26
27         cmp al,'A'
28         ja alphabet ;above 'A', there's no chance of having a digit
29         jb digit    ;below 'A', there's no chance of having a alphabet
30
31
32     alphabet:
33         inc cx
34         jmp l1
35
36     digit:
37         cmp al,20h ; for avoiding space
38         jz l1
39         inc dx
40         jmp l1
41
42         ; for showing output
43     end:
44         mov ax,cx
45         CALL PTHIS
46         db 13, 10, 'Alphabet : ', 0
47         CALL PRINT_NUM
48
49         mov ax,dx
50         CALL PTHIS
51         db 13, 10, 'Digit : ', 0
52         CALL PRINT_NUM
53
54         DEFINE_PRINT_NUM
55         DEFINE_PRINT_NUM_UN
56         DEFINE_PTHIS
57
58     hlt
59
60
61 endp main
62 end main

```

```

1  include "emu8086.inc"
2  .model small
3  .stack 100h
4
5  .data
6      inputArray db 15,3,2,2,1,6,12
7      outputArray db 6 dup(?)
8  .code
9
10 main proc
11     mov ax,@data
12     mov ds,ax
13
14     CALL SCAN_NUM ;CX contains range input
15
16     ;initializaion
17     mov si,0
18     mov bh,1
19     mov di,0
20     inc cl
21     mov ah,0
22
23     l1:
24         cmp bh,cl
25         jz mid
26         mov al,inputArray[si]
27         cmp al,bh
28         jz found           ;if found,then straight start with new search
29         inc si
30         cmp si,7
31         jz notFound        ;if not found, then append
32                             ;it into the outputArray
33         jmp l1
34
35     found:
36         mov si,0
37         inc bh
38         jmp l1
39
40     notFound:
41         mov outputArray[di],bh
42         inc di
43         jmp l1
44
45     mid:           ;this segment is to show the output
46         mov outputArray[di],'$'
47         mov di,0
48
49         CALL PTHIS
50         db 13, 10, 'Missing values are : ', 0
51     l2:
52         mov al,outputArray[di]
53         cmp al,'$'
54         jz end
55         CALL PTHIS
56         db ' ', 0
57         CALL PRINT_NUM
58         inc di
59         jmp l2
60
61     end:
62         DEFINE_SCAN_NUM
63         DEFINE_PRINT_NUM
64         DEFINE_PRINT_NUM_UN
65         DEFINE_PTHIS
66         hlt
67
68     endp main
69     end main

```