

Lab # 10

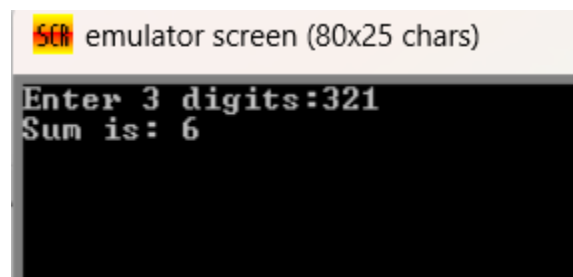
Array Processing in x86 Assembly Language

EXERCISE:

Q1) Calculate and display the sum of three numbers entered by the user, save input in an array.

```
01 .model small
02 .stack 100h
03 .data
04     nums db 3 dup(?) ; array to store 3 input digits
05     msg1 db 'Enter 3 digits:$'
06     msg2 db 13,10,'Sum is: $'
07     sum db 0
08
09 .code
10 main:
11     mov ax, @data
12     mov ds, ax
13
14     lea dx, msg1
15     mov ah, 9
16     int 21h
17
18     mov si, 0
19
20 input_loop:
21     mov ah, 1
22     int 21h
23     sub al, '0'
24     mov nums[si], al
25     inc si
26     cmp si, 3
27     jne input_loop
28
29     mov al, nums[0]
30     add al, nums[1]
31     add al, nums[2]
32     mov sum, al
33
34     lea dx, msg2
35     mov ah, 9
36     int 21h
37
38     mov al, sum
39     add al, '0'
40     mov dl, al
41     mov ah, 2
42     int 21h
43
44     mov ah, 4ch
45     int 21h
46 end main
```

OUTPUT:



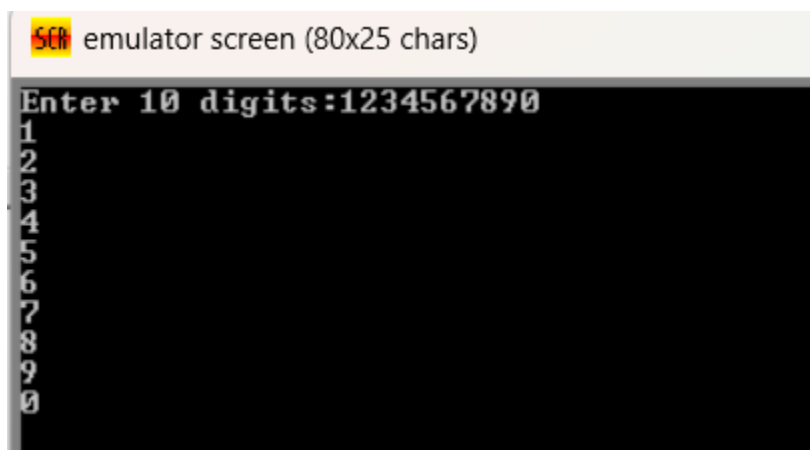
emulator screen (80x25 chars)

```
Enter 3 digits:321
Sum is: 6
```

Q2) Enter 10 elements in an array using the input function and display them in a new line.

```
01 .model small
02 .stack 100h
03 .data
04     arr db 10 dup(?)
05     prompt db 'Enter 10 digits:$'
06     newline db 13,10,'$'
07
08 .code
09 main:
10     mov ax, @data
11     mov ds, ax
12
13     lea dx, prompt
14     mov ah, 9
15     int 21h
16
17     mov si, 0
18
19 read_loop:
20     mov ah, 1
21     int 21h
22     sub al, '0'
23     mov arr[si], al
24     inc si
25     cmp si, 10
26     jne read_loop
27
28     lea dx, newline
29     mov ah, 9
30     int 21h
31
32     mov si, 0
33
34 print_loop:
35     mov dl, arr[si]
36     add dl, '0'
37     mov ah, 2
38     int 21h
39
40     lea dx, newline
41     mov ah, 9
42     int 21h
43
44     inc si
45     cmp si, 10
46     jne print_loop
47
48     mov ah, 4ch
49     int 21h
50 end main
51
```

OUTPUT:



emulator screen (80x25 chars)

```
Enter 10 digits:1234567890
1
2
3
4
5
6
7
8
9
0
```

Q3)write a code to Save your name in an array and Display in reverse order .

```
01 .model small
02 .stack 100h
03 .data
04     myname db 'marium'
05     length db 6
06     newline db 13,10,'$'
07
08 .code
09 main:
10     mov ax, @data
11     mov ds, ax
12
13     lea dx, newline
14     mov ah, 9
15     int 21h
16
17     mov cx, 0
18     mov cl, length
19
20     mov si, offset myname
21     add si, cx
22     dec si
23
24 print_loop:
25
26     mov dl, [si]
27     mov ah, 2
28     int 21h
29
30     dec si
31     loop print_loop
32
33     lea dx, newline
34     mov ah, 9
35     int 21h
36
37     mov ah, 4ch
38     int 21h
39 end main
40
```

OUTPUT:

