

Computer Organization and Assembly Language (EL 2003)



Lab 4

Lab # 3 Last Question Solution

$$x = 5$$

$$y = 10$$

$$A = x + y - 3$$

$$B = A - 9 + y$$

```
01 .model small
02 .stack 100h
03 .data
04     x db 5
05     y db 10
06     A db ?
07     B db ?
08 .code
09     mov ax, @data
10     mov ds, ax
11
12     mov ax, 0
13     mov al, x
14     add al, y
15     sub al, 3
16     mov A, al
17
18     sub al, 9
19     add al, y
20     mov B, al
21
22     mov ah, 4ch
23     int 21h
```

variables	
size:	byte
elements:	1
edit	show as: hex
X	05h
Y	0Ah
A	0Ch
B	0Dh

Outline

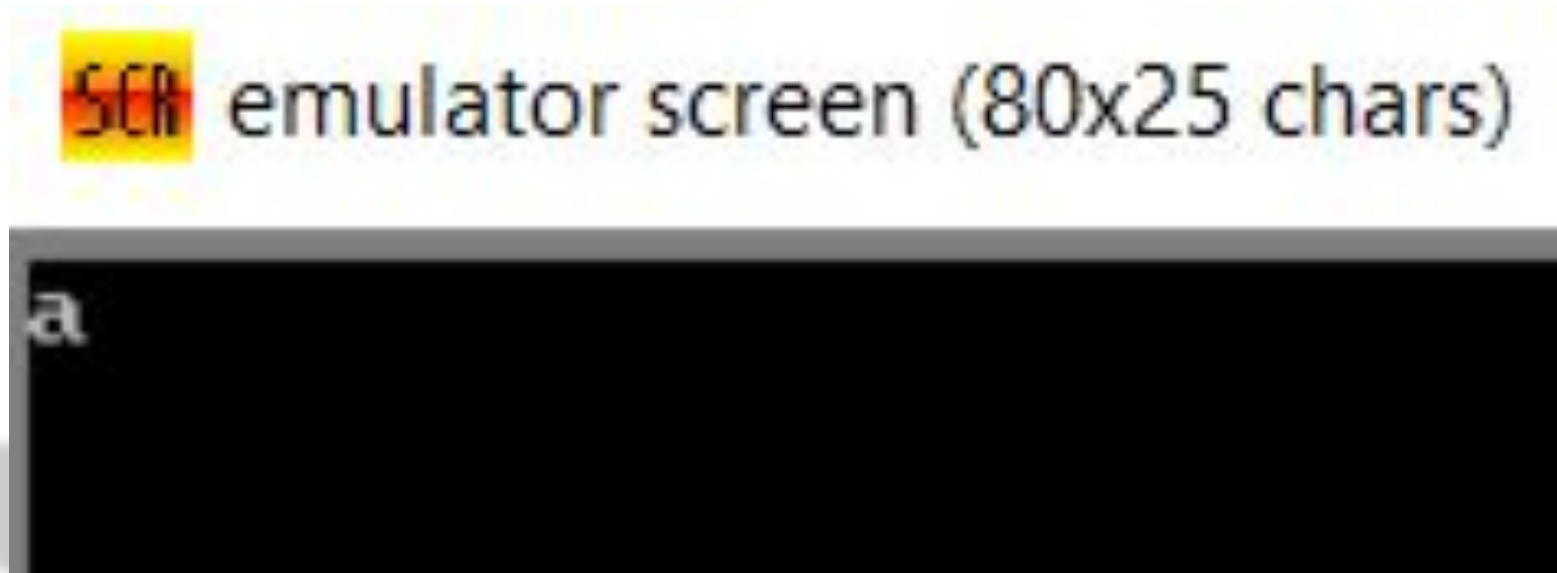
- Input Output Operations
 - INT 21H
 - Function Calls: 01H, 02H, 09H, 0AH

Brief Function Calls Usage

AH Register Value of Function Call No.	Output	Functionality
MOV AH, 02H	On Screen	Display the content of register DL on screen in ASCII form.
MOV AH, 09H	On Screen	Display the string that is terminated by"\$" sign.
MOV AH, 01H	AL Register	Reads a character from keyboard, stores it in AL and display it (echo it) on screen.
MOV AH, 08H	Al Register	Read character from keyboard without echoing it on screen
MOV AH, 0AH	Offset in DX	Read a string of characters from keyboard.

Displaying A Character on Screen

```
01 .model small
02 .stack 100h
03 .data
04 .code
05     mov ax, @data
06     mov ds, ax
07
08     mov ah, 02h
09     mov dl, 'a'
10     int 21h
11
12 mov ah, 4ch
13 int 21h
```



**Display the content of register DL
on screen in ASCII form.**

Displaying A String on Screen

String terminated by ``\$'' sign.

```
01 .model small
02 .stack 100h
03 .data
04 hello db 'Hello World!', '$'
05 .code
06     mov ax, @data
07     mov ds, ax
08
09     mov ah, 09h
10     lea dx, hello
11     int 21h
12
13 mov ah, 4ch
14 int 21h
```

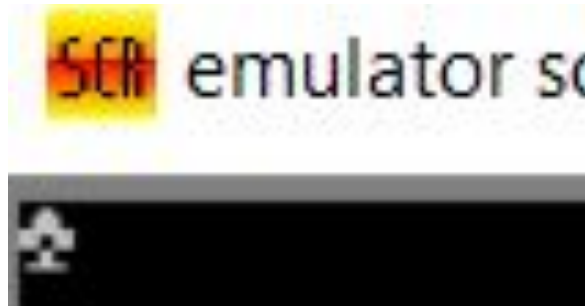
SCR emulator screen (80x25 chars)

Hello World!

Display the string that is terminated by ``\$'' sign.

Displaying a Number on Screen

```
01 .model small
02 .stack 100h
03 .data
04 .code
05     mov ax, @data
06     mov ds, ax
07
08     mov bl, 5
09     mov ah, 02h
10     mov dl, bl
11     int 21h
12
13     mov ah, 4ch
14     int 21h
```



Why 5 is not displayed?


**Because it displays the
ASCII Value**

**To display the number,
ADD 30h in it.**

00:	null	20:	spa
01:	☺	21:	!
02:	☹	22:	"
03:	♥	23:	#
04:	♠	24:	\$
05:	♣	25:	%
06:	♣	26:	&
07:	beep	27:	'
08:	back	28:	<
09:	tab	29:	>
0A:	newl	2A:	*
0B:	♂	2B:	+
0C:	♀	2C:	,
0D:	cret	2D:	-
0E:	☾	2E:	.
0F:	☼	2F:	/
10:	▶	30:	0
11:	◀	31:	1
12:	↕	32:	2
13:	!!	33:	3
14:	99	34:	4
15:	88	35:	5
16:	-	36:	6

Displaying a Number on Screen

```
mov bl, 5  
add bl, 30H  
mov ah, 02h  
mov dl, bl  
int 21h
```

 SCA emulator

5

Reading a Character from Console

```
mov ah, 01h  
int 21h
```

registers		SCSI emulator	
	H	L	
AX	01	35	5

registers		SCSI emulator	
	H	L	
AX	01	61	a

ASCII Values in AL
Register

Reading a Character from Console

```
mov ah, 08h  
int 21h
```

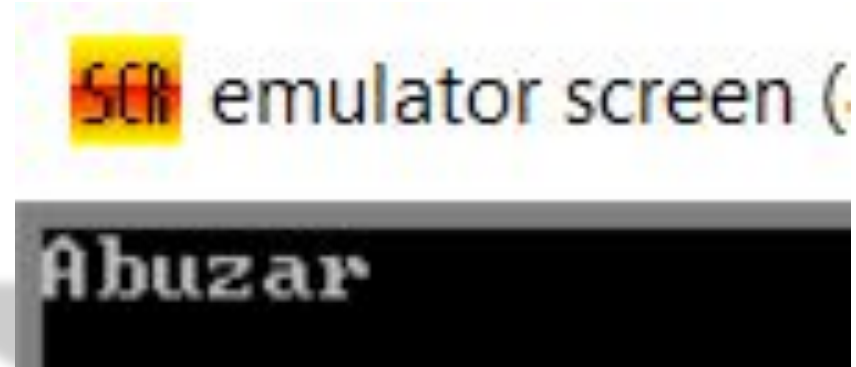
registers		SCN emulator sc	
	H	L	
AX	08	35	

registers		SCN emulator scre	
	H	L	
AX	08	61	

Without Echoing Reads
ASCII Values in AL
Register

Reading a String From Console

```
01 .model small
02 .stack 100h
03 .data
04     username db 19, 20 DUP(0)
05 .code
06     mov ax, @data
07     mov ds, ax
08
09     mov ah, 0Ah
10     lea dx, username
11     int 21h
```



Total Size of string
variable

Reading a String From Console

The screenshot shows a debugger's 'variables' window with three instances of a variable named 'USERNAME'. Each instance has a 'size' of 'byte' and 'elements' of '20'. The first instance is set to 'show as: ascii' and displays characters like '!', ' ', 'A', 'b', 'u', 'z', 'a', 'r', 'P', and 'NU'. The second instance is set to 'show as: signed' and displays numerical values: 19, 6, 65, 98, 115, 122, 97, and so on. The third instance is set to 'show as: hex' and displays hexadecimal values: 13h, 06h, 41h, 62h, 75h, 7Ah, 61h, 72h, 0Dh, and 00h. Red arrows originate from two yellow callout boxes. One arrow points from the 'elements: 20' field of the top instance to the first two data cells (19 and 6) of the middle instance. The other arrow points from the 'elements: 20' field of the middle instance to the same two data cells (19 and 6).

Instance	Size	Elements	Show as	Value
1	byte	20	ascii	'!', ' ', 'A', 'b', 'u', 'z', 'a', 'r', 'P', 'NU', ...
2	byte	20	signed	19, 6, 65, 98, 115, 122, 97, ...
3	byte	20	hex	13h, 06h, 41h, 62h, 75h, 7Ah, 61h, 72h, 0Dh, 00h, ...

Total Size of string variable

Actual size of variable, no. of characters entered.

Reading a String From Console

- The variable should be at least three bytes longer than the largest input string anticipated.



First Byte Reserved for Total Size.

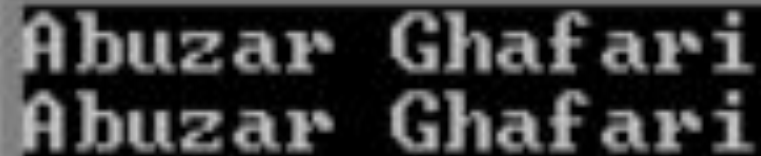
Second Byte Reserved for Actual String Size.

Last Byte Reserved for to terminate with \$

Read and Display Name

```
01 .model small
02 .stack 100h
03 .data
04     username db 19, 20 DUP(0)
05     crlf db 0dh, 0ah, '$'
06 .code
07     mov ax, @data
08     mov ds, ax
09
10     ; Read Name
11     mov ah, 0Ah
12     lea dx, username
13     int 21h
14
15     ; End Line
16     mov ah, 09h
17     lea dx, crlf
18     int 21h
```


Read and Display Name



Abuzar Ghafari
Abuzar Ghafari

```
19
20     ; Insert $ to the end of the name,
21     ; Get the address of the starting index of the name, i.e. 2nd index
22     mov bx, 0; initialize bx to 0.
23     mov bl, username[1]; contains, how many characters entered
24     mov username[bx+2], '$' ; terminate the string with $ symbol
25     mov ah, 09h
26     lea dx, username[2] ; starting address of the name
27     int 21h
28
29
30     mov ah, 4ch
31     int 21h
```




Thank you!