Computer Organization and Assembly Language (EL 2003)

Lab 4

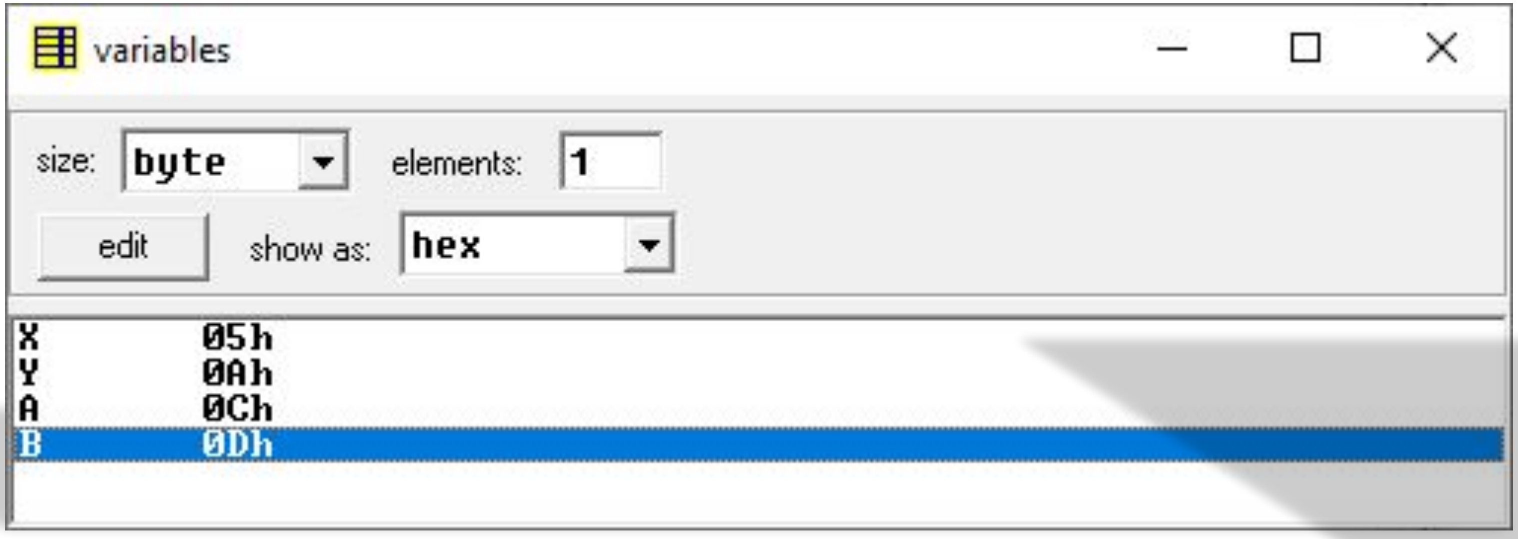
Lab # 3 Last Question Solution

$$x = 5$$

$$y = 10$$

$$A = x + y - 3$$

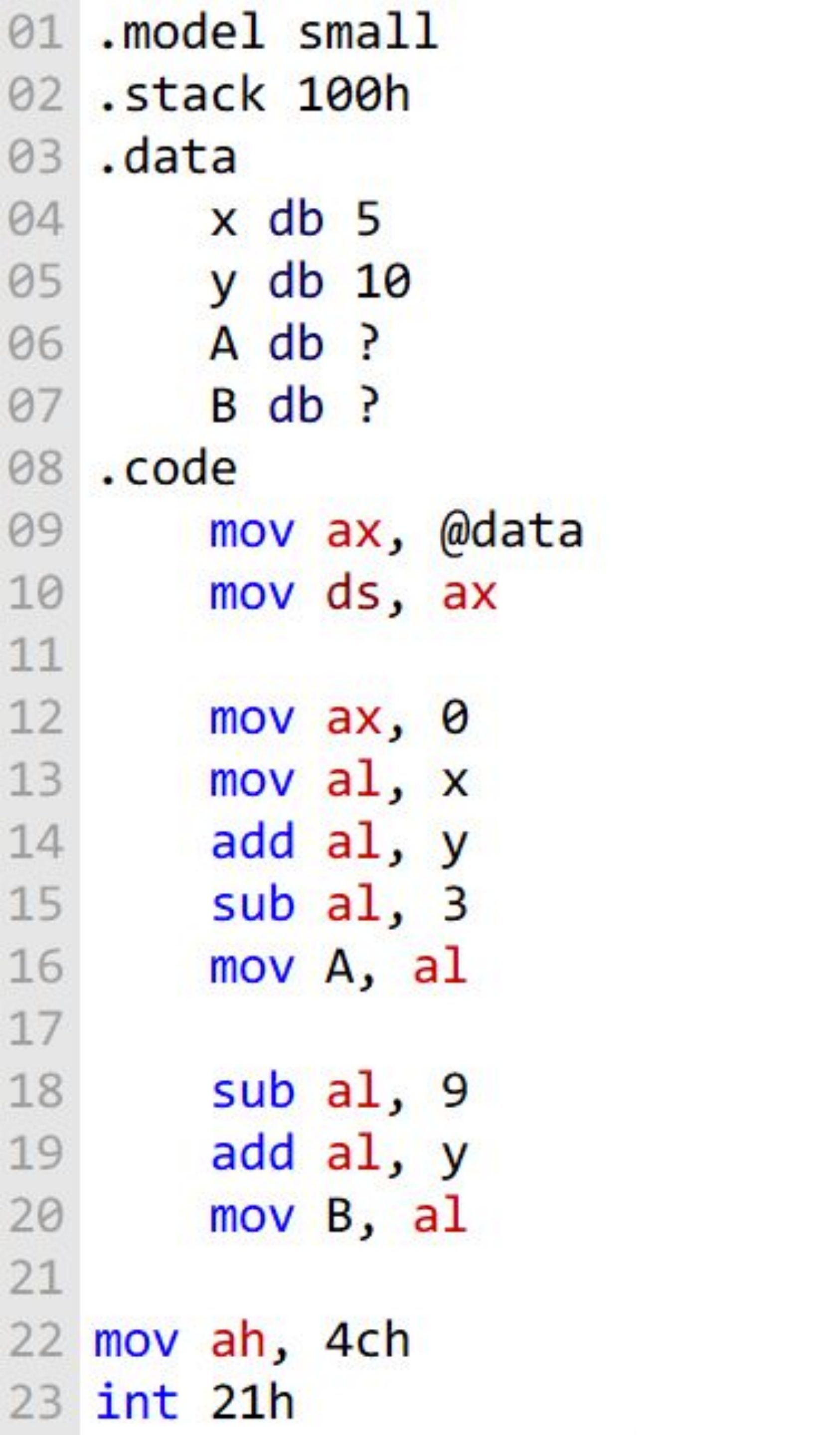
$$B = A - 9 + y$$



size: byte elements: 1

show as: hex

```
X        05h
Y        0Ah
A        0Ch
B        0Dh
```



```
.model small
.stack 100h
.data
    x db 5
    y db 10
    A db ?
    B db ?
.code
    mov ax, @data
    mov ds, ax

    mov ax, 0
    mov al, x
    add al, y
    sub al, 3
    mov A, al

    sub al, 9
    add al, y
    mov B, al

mov ah, 4ch
int 21h
```

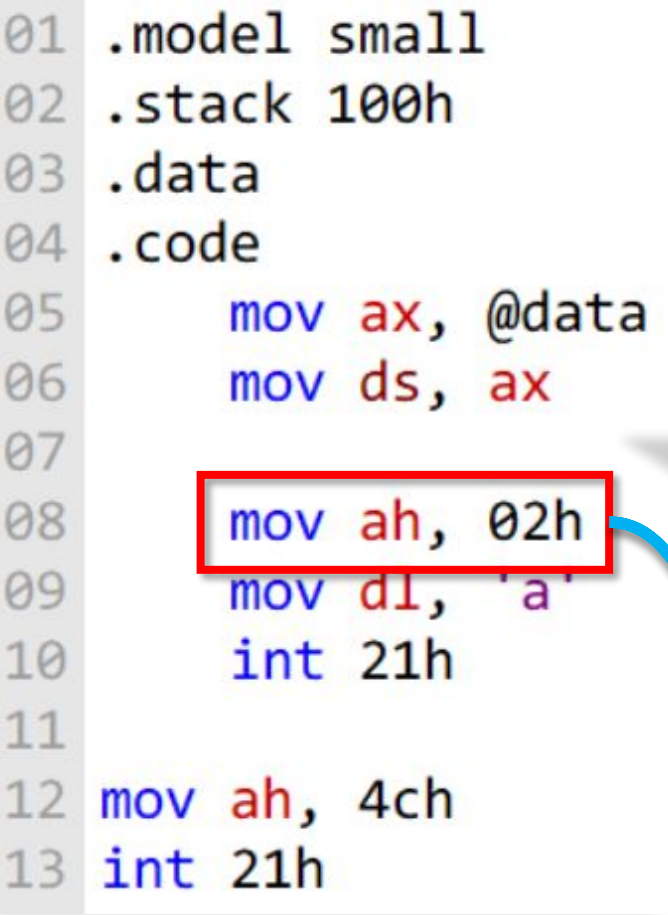
Outline

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

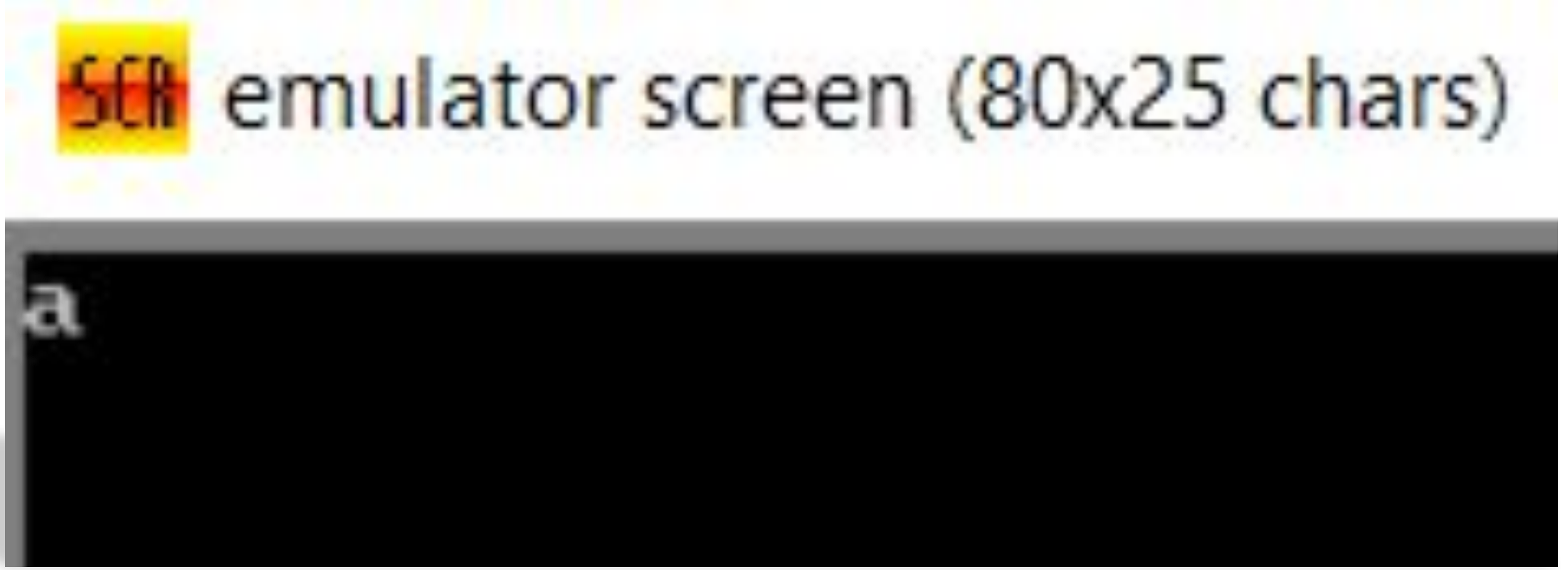
Brief Function Calls Usage

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



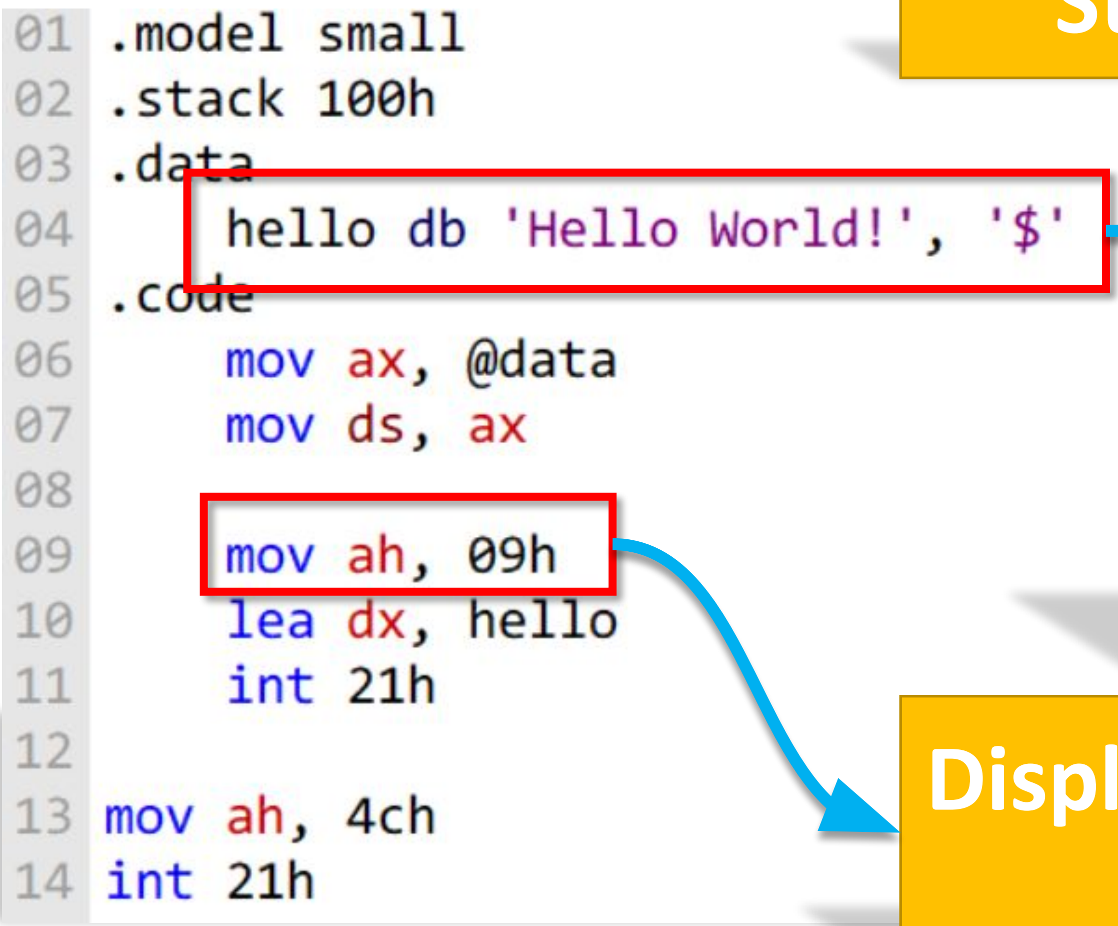
```
.model small
.stack 100h
.data
.code
    mov ax, @data
    mov ds, ax

    mov ah, 02h
    mov dl, 'a'
    int 21h

mov ah, 4ch
int 21h
```



```
a
```

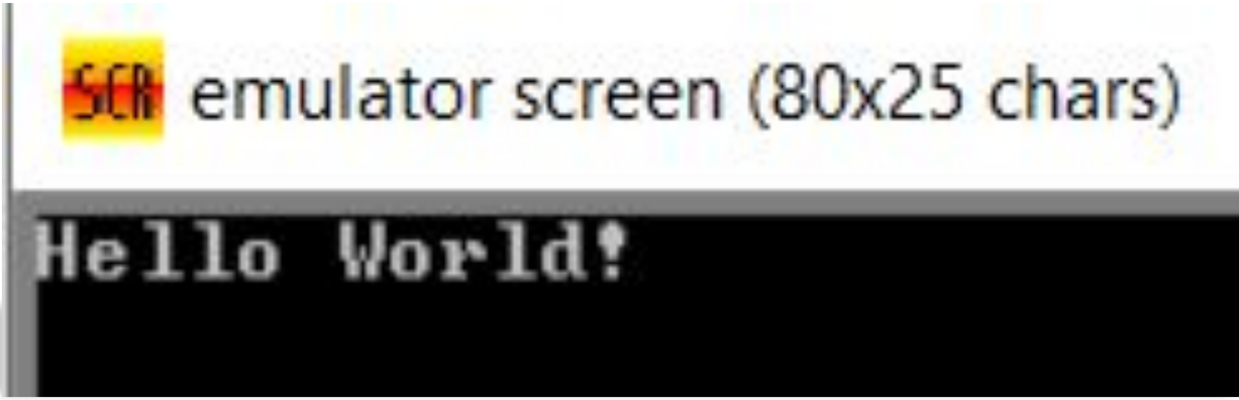
Display the content of register DL on screen in ASCII form.

Displaying A String on Screen



```
.model small
.stack 100h
.data
    hello db 'Hello World!', '$'
.code
    mov ax, @data
    mov ds, ax

    mov ah, 09h
    lea dx, hello
    int 21h

mov ah, 4ch
int 21h
```

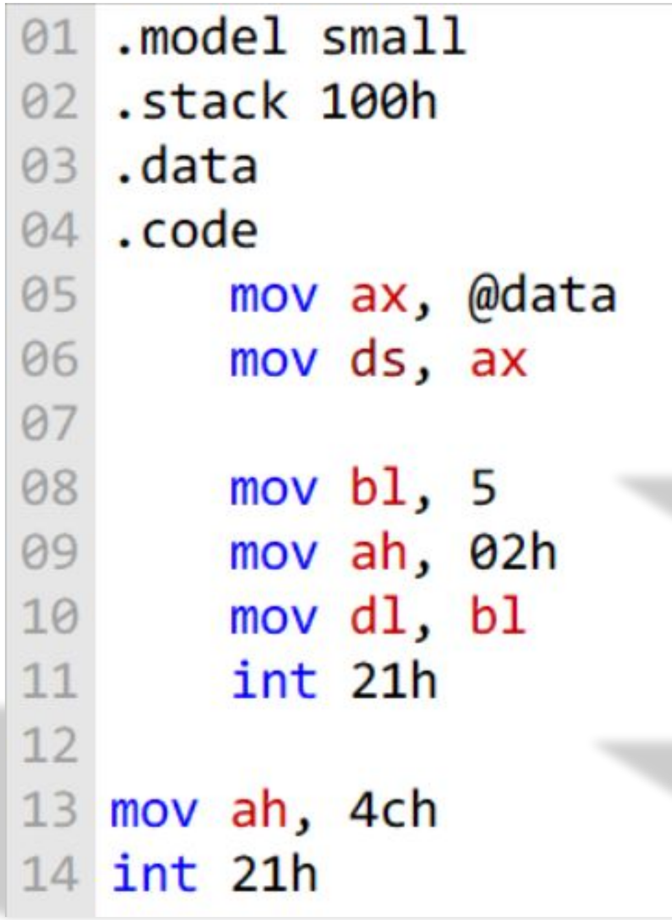
String terminated by``$" sign.



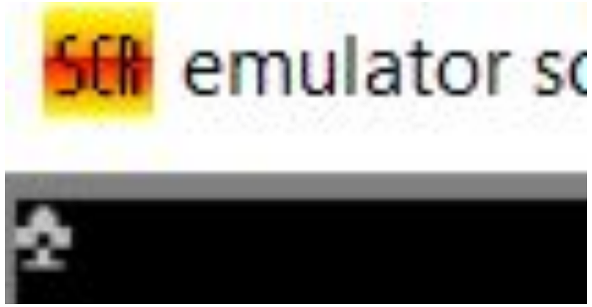
```
Hello World!
```

Display the string that is terminated by``$" sign.

Displaying a Number on Screen



```
.model small
.stack 100h
.data
.code
        mov ax, @data
        mov ds, ax

        mov bl, 5
        mov ah, 02h
        mov dl, bl
        int 21h

mov ah, 4ch
int 21h
```

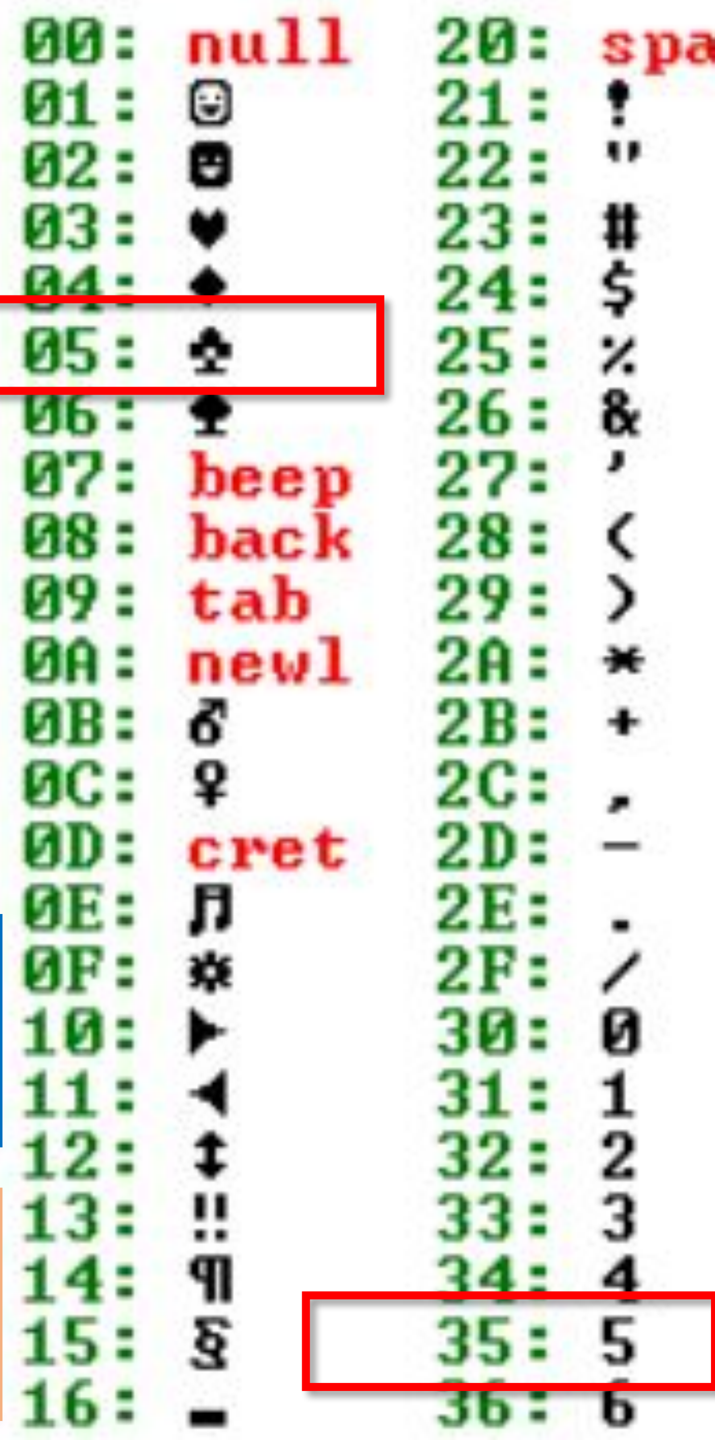


```
♣
```

Why 5 is not displayed?

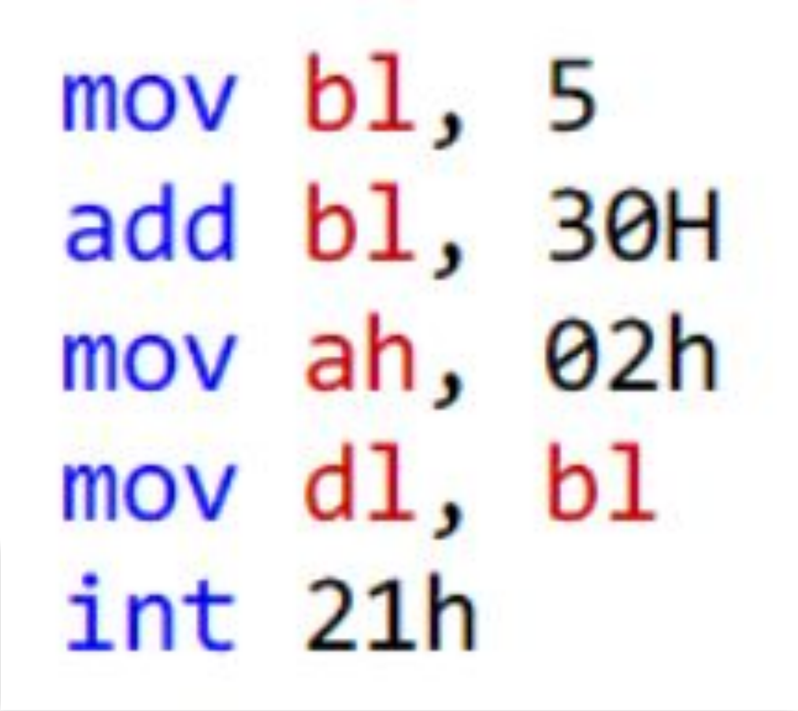
Because it displays the ASCII Value

To display the number, ADD 30h in it.

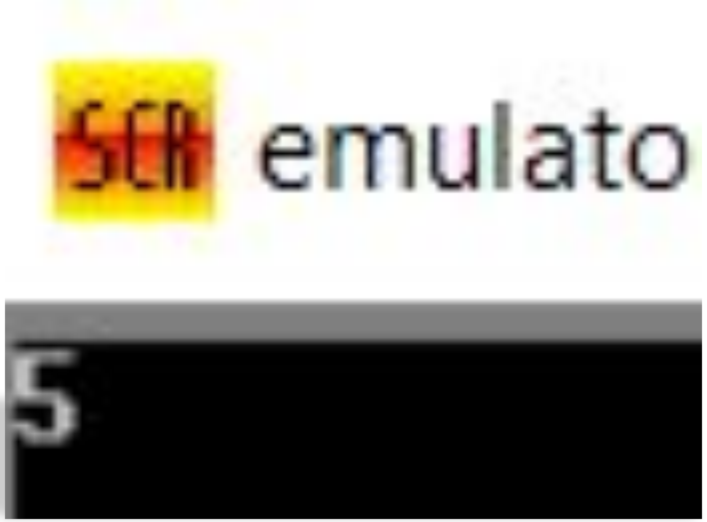


00:	null	20:	spa
01:	☺	21:	!
02:	☻	22:	"
03:	♥	23:	#
04:	♦	24:	$
05:	♣	25:	%
06:	♠	26:	&
07:	beep	27:	'
08:	back	28:	(
09:	tab	29:	)
0A:	newl	2A:	*
0B:	♂	2B:	+
0C:	♀	2C:	,
0D:	cret	2D:	-
0E:	♫	2E:	.
0F:	☼	2F:	/
10:	►	30:	0
11:	◄	31:	1
12:	↕	32:	2
13:	‼	33:	3
14:	¶	34:	4
15:	§	35:	5
16:	▬	36:	6

Displaying a Number on Screen

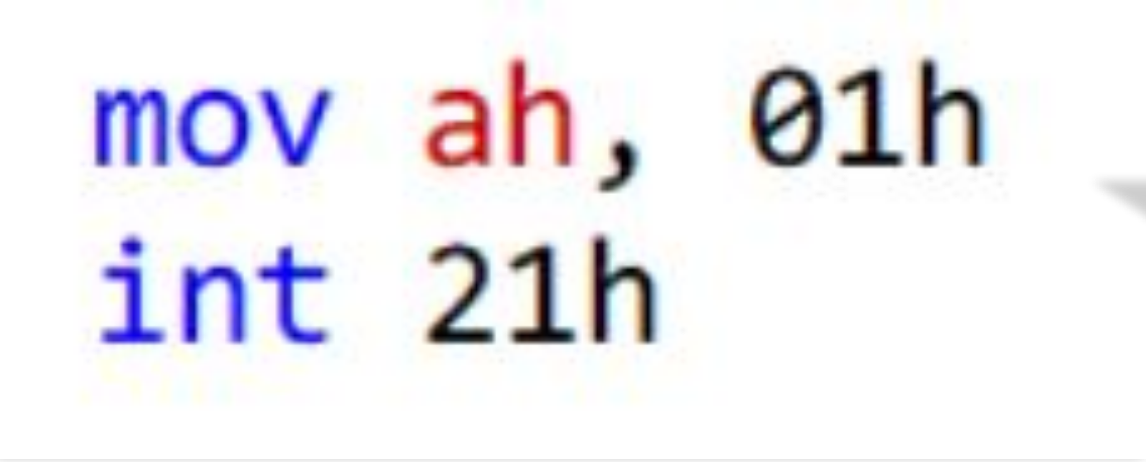


```
mov bl, 5
add bl, 30H
mov ah, 02h
mov dl, bl
int 21h
```

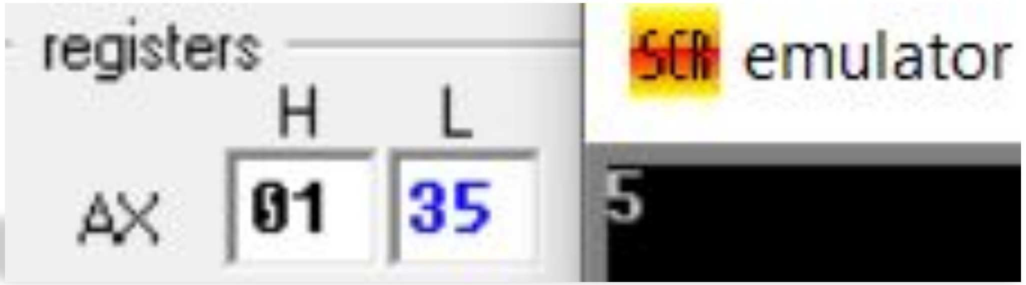


```
5
```

Reading a Character from Console

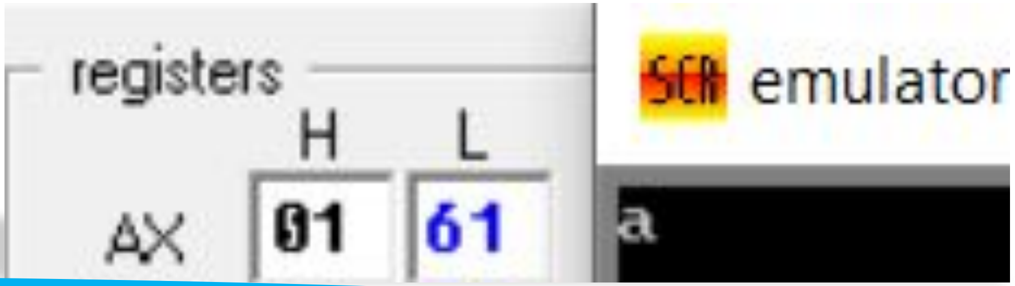


```
mov ah, 01h
int 21h
```



```
     H    L
AX  01   35
```

```
5
```

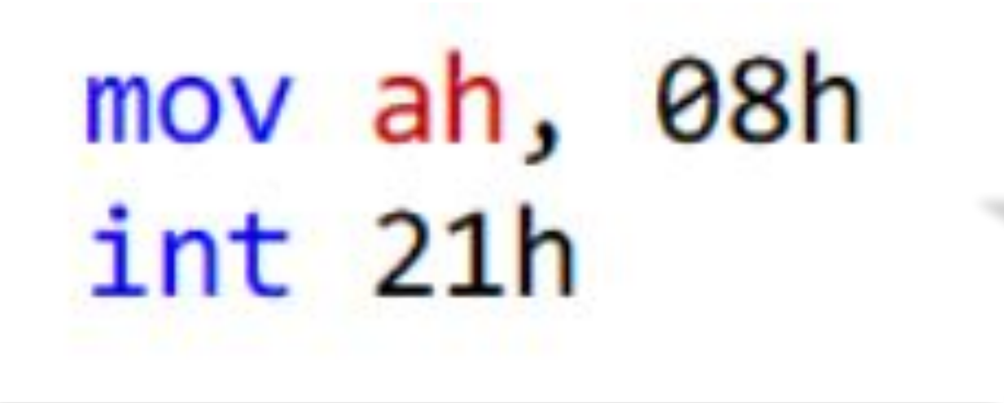


```
     H    L
AX  01   61
```

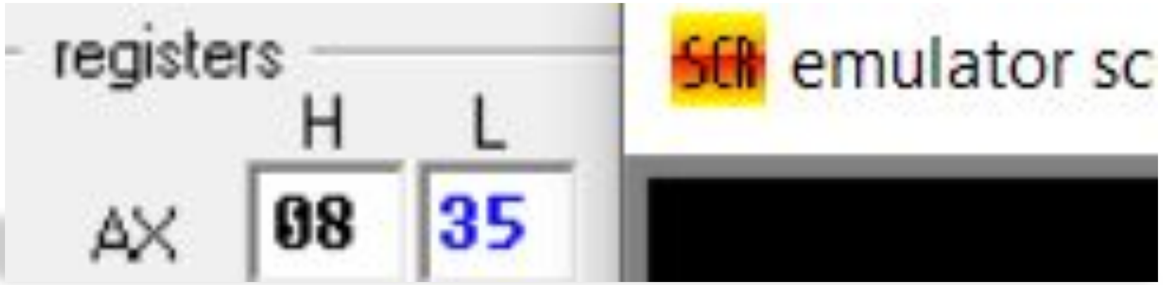
```
a
```

ASCII Values in AL Register

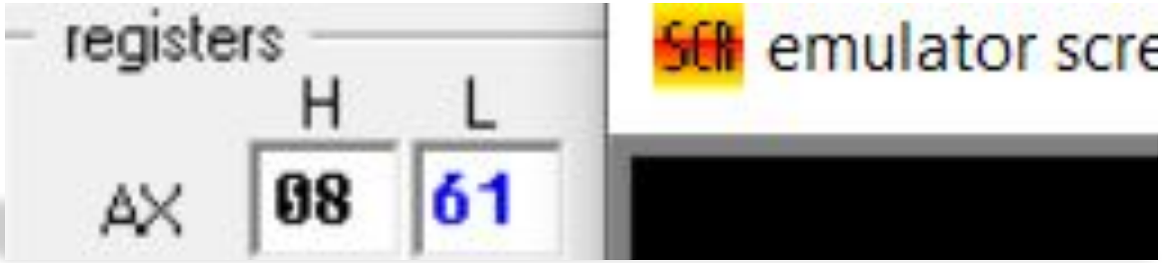
Reading a Character from Console



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



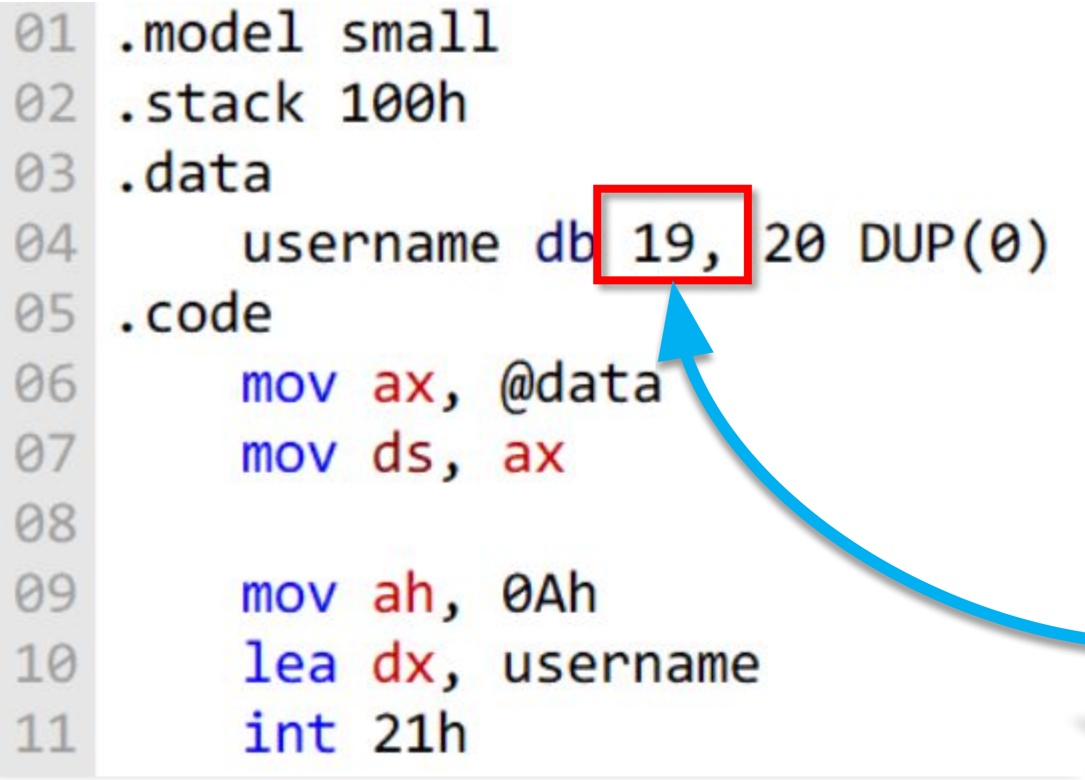
```
registers
        H    L
AX     08   35
```



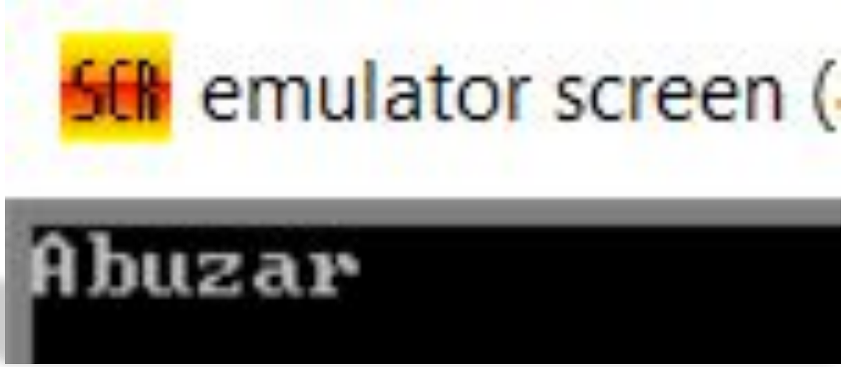
```
registers
        H    L
AX     08   61
```

Without Echoing Reads ASCII Values in AL Register

Reading a String From Console



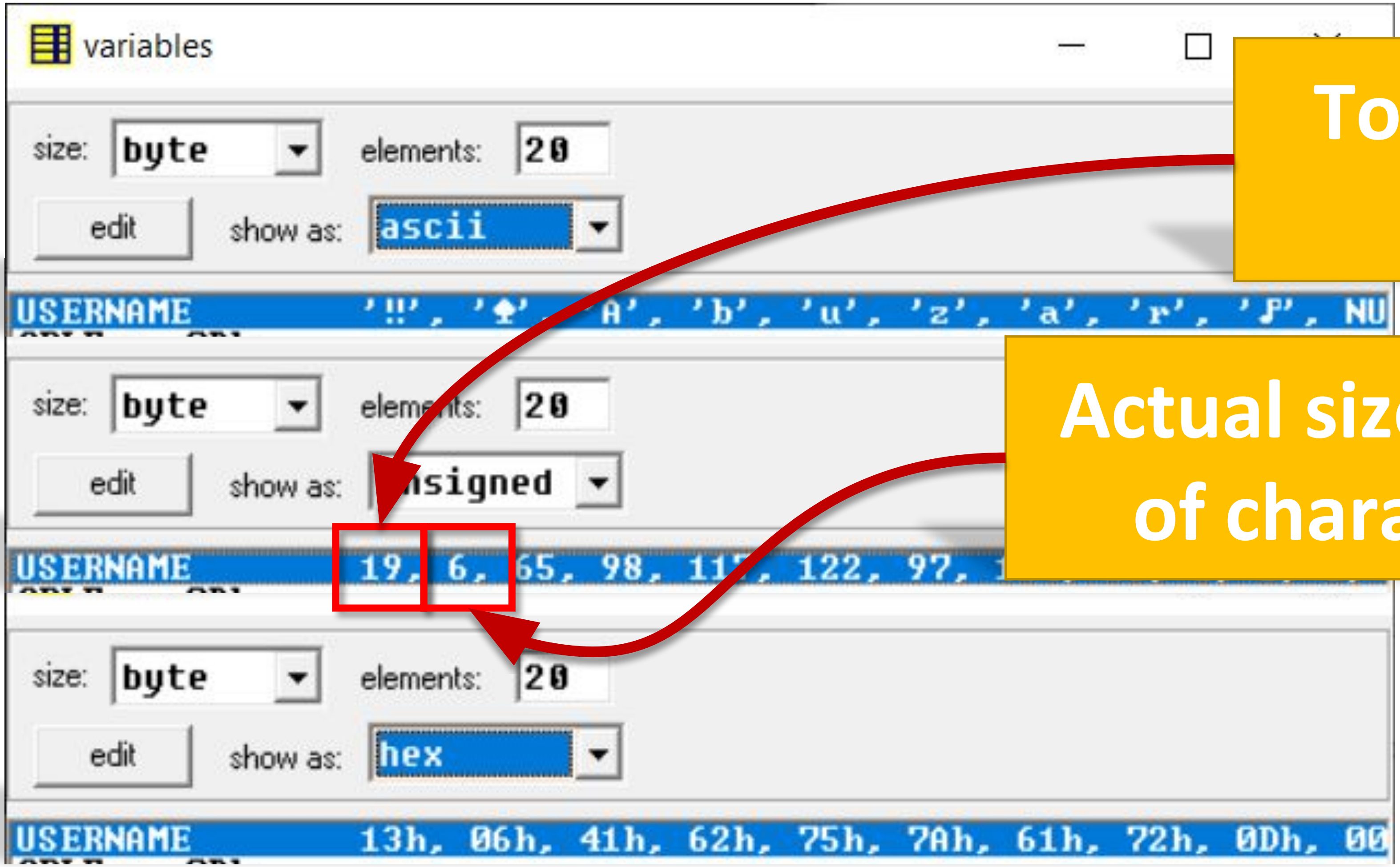
```
.model small
.stack 100h
.data
    username db 19, 20 DUP(0)
.code
    mov ax, @data
    mov ds, ax

    mov ah, 0Ah
    lea dx, username
    int 21h
```



```
Abuzar
```

Total Size of string variable

Reading a String From Console



size: byte elements: 20
edit show as: ascii

```
USERNAME          '‼', '♠', 'A', 'b', 'u', 'z', 'a', 'r', '♪', NU
```

size: byte elements: 20
edit show as: unsigned

```
USERNAME          19, 6, 65, 98, 117, 122, 97,
```

size: byte elements: 20
edit show as: hex

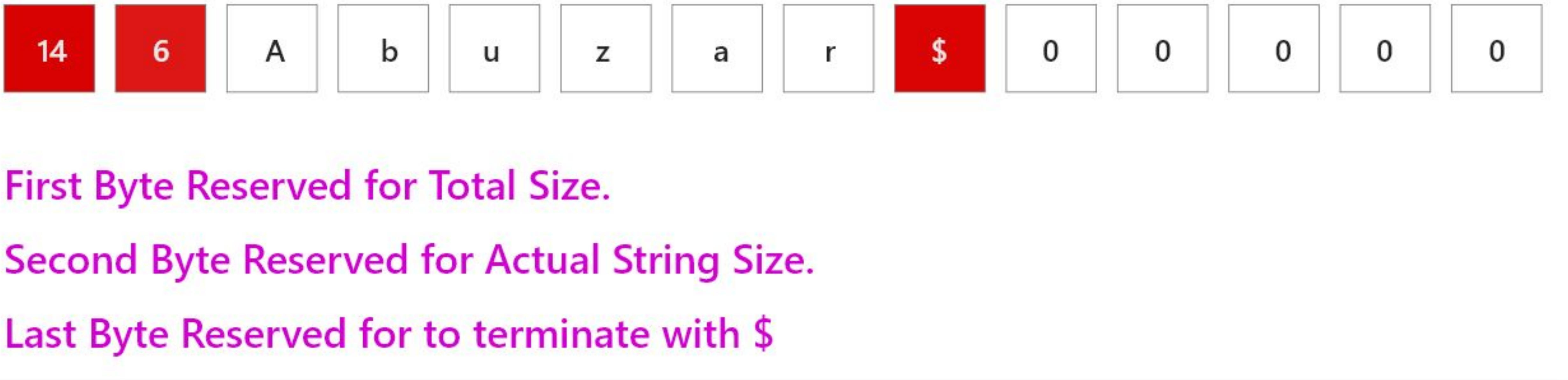
```
USERNAME          13h, 06h, 41h, 62h, 75h, 7Ah, 61h, 72h, 0Dh, 00
```

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.



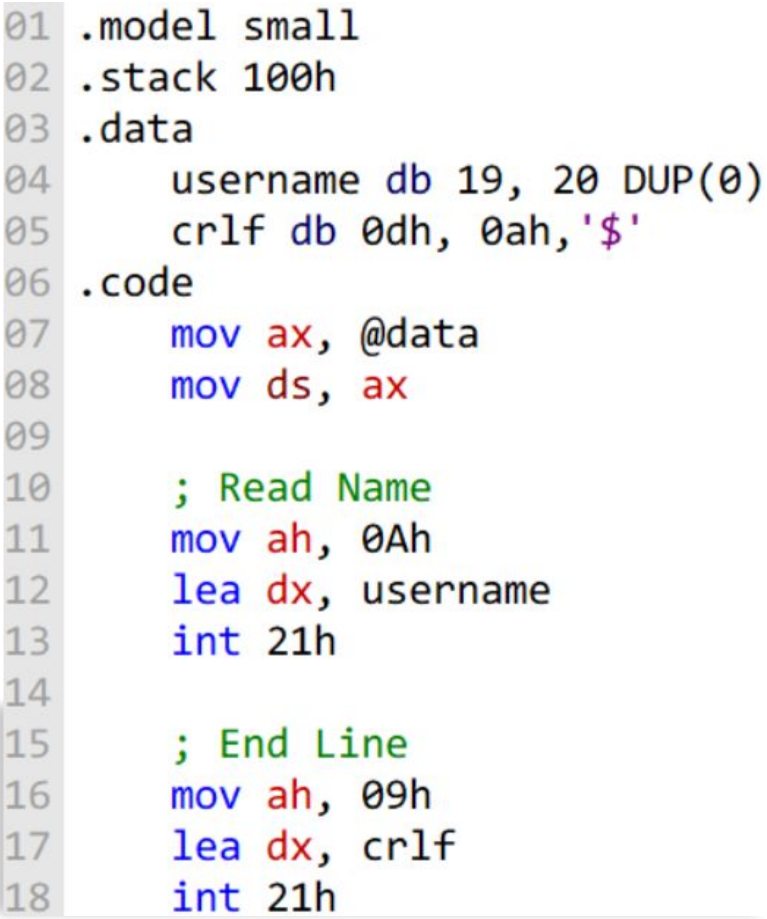
14	6	A	b	u	z	a	r	$	0	0	0	0	0

First Byte Reserved for Total Size.

Second Byte Reserved for Actual String Size.

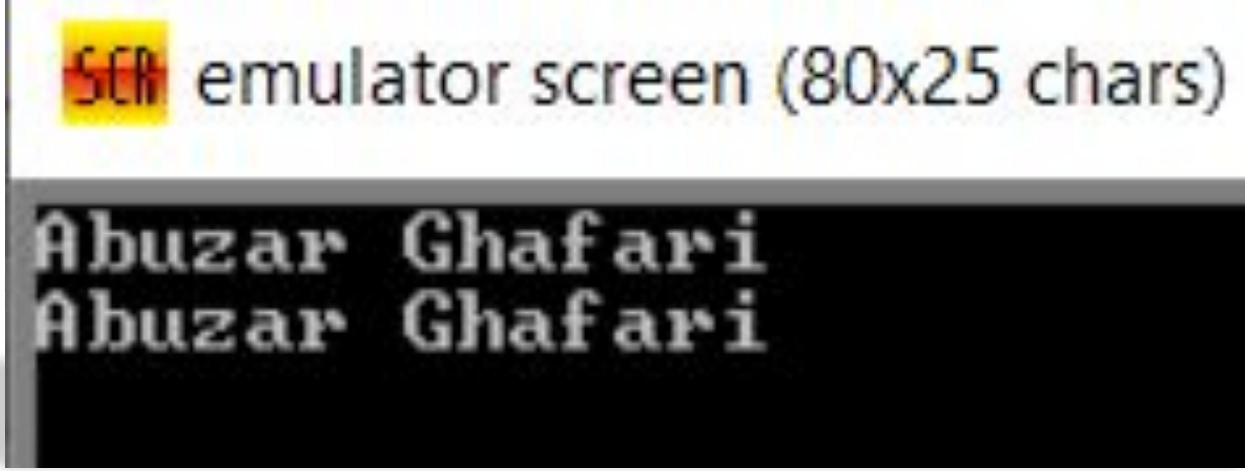
Last Byte Reserved for to terminate with $

Read and Display Name

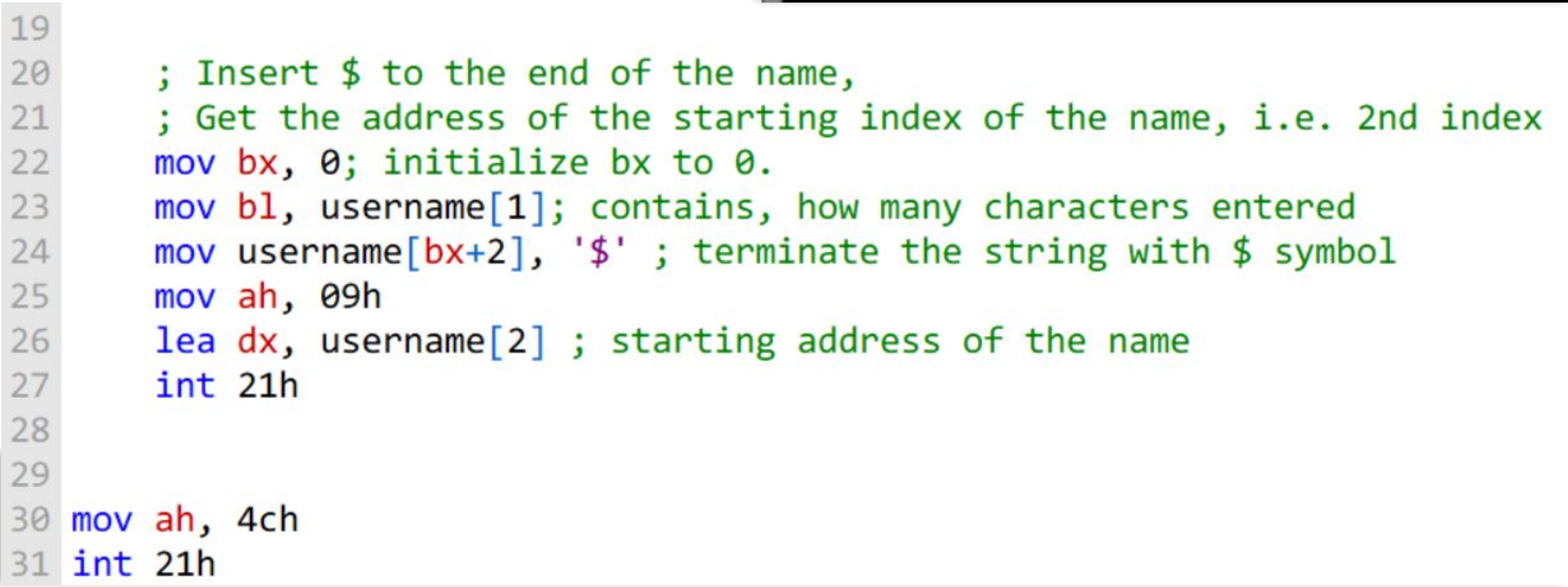


```
.model small
.stack 100h
.data
    username db 19, 20 DUP(0)
    crlf db 0dh, 0ah,'$'
.code
    mov ax, @data
    mov ds, ax

    ; Read Name
    mov ah, 0Ah
    lea dx, username
    int 21h

    ; End Line
    mov ah, 09h
    lea dx, crlf
    int 21h
```

Read and Display Name



```
Abuzar Ghafari
Abuzar Ghafari
```



```

    ; Insert $ to the end of the name,
    ; Get the address of the starting index of the name, i.e. 2nd index
    mov bx, 0; initialize bx to 0.
    mov bl, username[1]; contains, how many characters entered
    mov username[bx+2], '$' ; terminate the string with $ symbol
    mov ah, 09h
    lea dx, username[2] ; starting address of the name
    int 21h


mov ah, 4ch
int 21h
```

Thank you!