# Lab #03
# Input and Output Instruction

## Objective

- Understand the working of INT 21H instruction
- Understand program having basic Input and Output Instruction

## Theory

The processor cannot access the peripheral devices (like keyboard or monitor) directly. Microprocessor use BIOS routines or DOS routines to access peripherals. The BIOS (Basic Input/Output System) routines are store in ROM and directly run the I/O ports where DOS routine use BIOS routine to perform operation that's why has less complex than BIOS routine.

The BIOS and DOS both use INT (Interrupt) instruction, the interrupt is actually done by number which determine nature of interrupt. The format of interrupt instruction is:

```
            INT    interrupt number
e.g.        INT    21H     ; Invoke DOS routine
            INT    16H     ; Invoke BIOS routine
```

Some INT 21H Function

| Function Number | Routine |
|---|---|
| 01H | Take input with echo |
| 02H | Display character |
| 07H | Take input without echo |
| 09H | Display string |
| 4CH | Control back to DOS |
| 0BH | Check standard input status |
| 2AH | Get system Date |
| 2CH | Get system Time |

When INT 21H execute the DOS first see the function number in AH register, so all the function number must be placed in AH register to execute.

## Function 01H (Single character input with echo)

After instruction executes, the AL register get the ASCII Code of the character and if non-character key (control key) is press then than AL register get 00H. The 01H function also displays the character on screen.

```
MOV    AH, 01H        ; Move input key function to AH register
INT    21H            ; Invoke DOS interrupt
```

## Function 07H (Single character input without echo)

This function works as same as the function of 01H but it not display the press character on screen.

## Function 02H (Single character output)

This function displays the signal character whose ASCII Code value is present in DL register.

```
MOV    DL,'A'         ; Move the ASCII code of A ( A = 41H) in DL register
MOV    AH, 02H        ; Move the single character output function into AH register
INT    21H            ; Invoke DOS interrupt
```

## Function 09H (Display string)

The function display the string which is terminated by $ sign present in Data segment register. Data Segment address is store in DS register and DX register store the offset address of the string to display.

```
.DATA
MESSAGE     DB      0AH, 0DH,"INDUS UNIVERSITY$"
.CODE
        MOV    AX,@DATA              ; Get address of DATA segment
        MOV    DS, AX                ; Move address of DATA segment into DS register
        MOV    DX, OFFSET MESSAGE ;Get offset address of MESSAGE
        MOV    AH, 09H               ;Move display string function into AH register
        INT    21H                   ; Invoke DOS interrupt
```

## Function 4CH (DOS exit function)

When the program terminate (end) the control must return back to the DOS. This is done by 4CH function.

```
MOV    AH, 4CH        ; Move return to DOS function into AH register
INT    21H            ; Invoke DOS interrupt
```

## Function 0BH (Check for Character available function)

This function is used to check any available character of keyboard. This function return two values in AL register, if it's 00H then no character available and if it's FFH then a character available.

```
MOV    AH, 0BH          ; Check for any Character available
INT    21H              ; Invoke DOS interrupt
```

## Function 2AH (Get system Date function)

This function returns the Date of the system. The return values are store in varies registers that are CX (year (1980-2099)), DH (month (00-11)), DL (day (0-30)) and AL (day of the week (00 for Sunday))

```
MOV    AH, 2CH          ; Get current time of the system
INT    21H              ; Invoke DOS interrupt
```

## Function 2CH (Get system Time function)

This function return the time of the system. The return values store in varies registers that are CH (hour (0-23)), CL (minutes (0-59)), DH (second (0-59)) and DL (1/100 second (0-99))

```
MOV    AH, 2CH          ; Get current time of the system
INT    21H              ; Invoke DOS interrupt
```

Hour store in CH register, Minutes store in CL register, Second store in DH register and milli-second store in DL register.

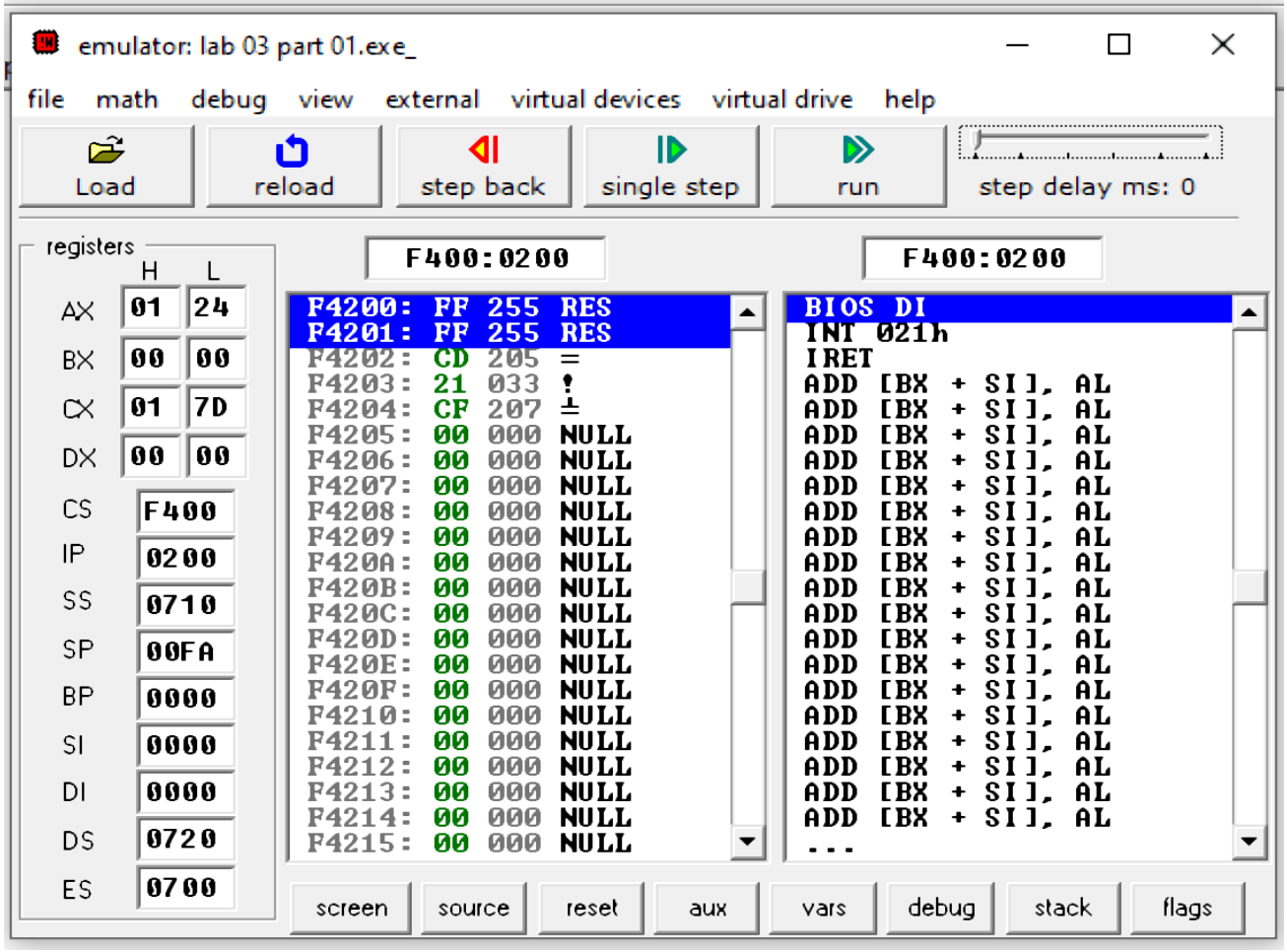
---

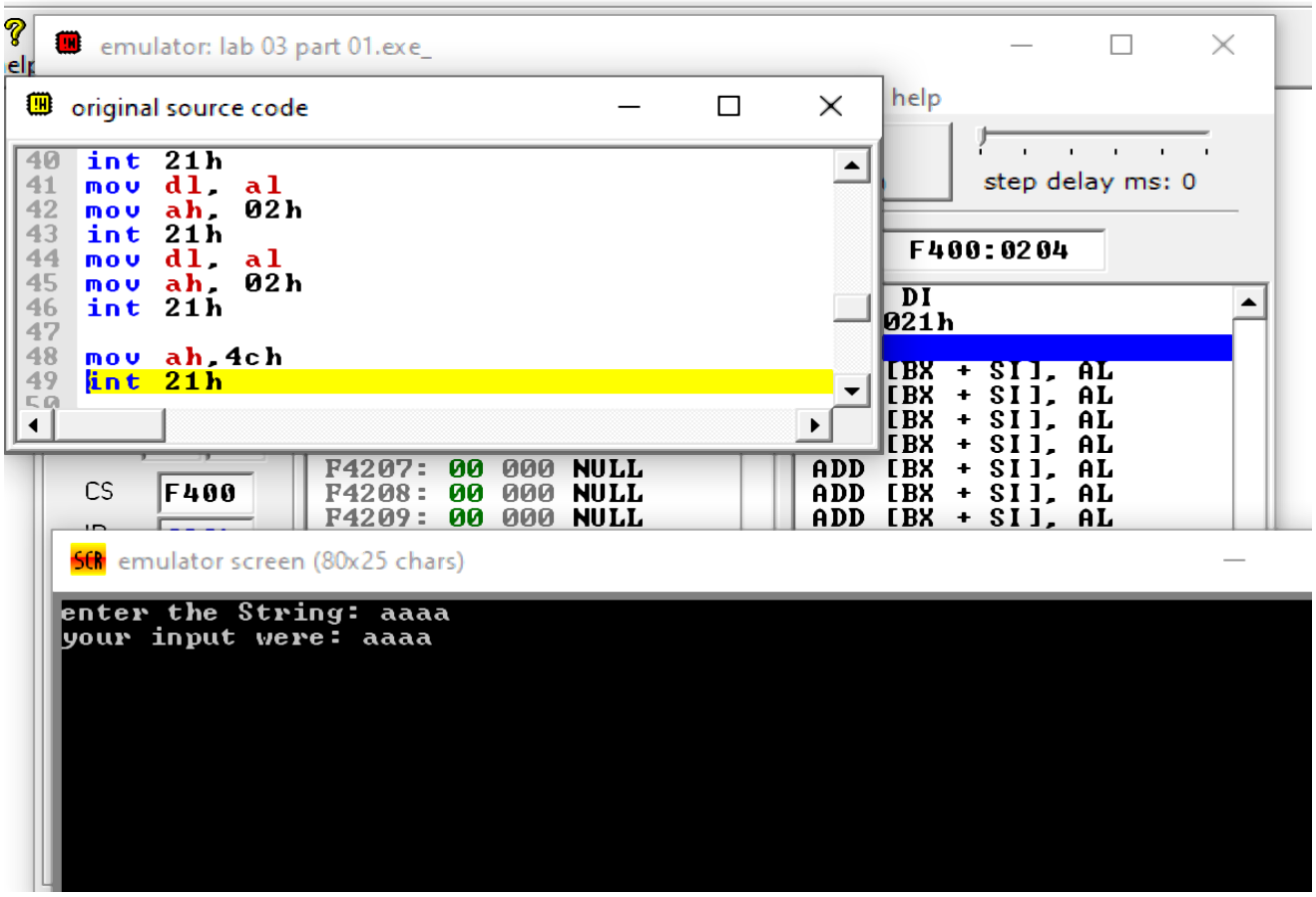
**Tasks:**

1- Write a program to take 4 character input from user and display user input on screen

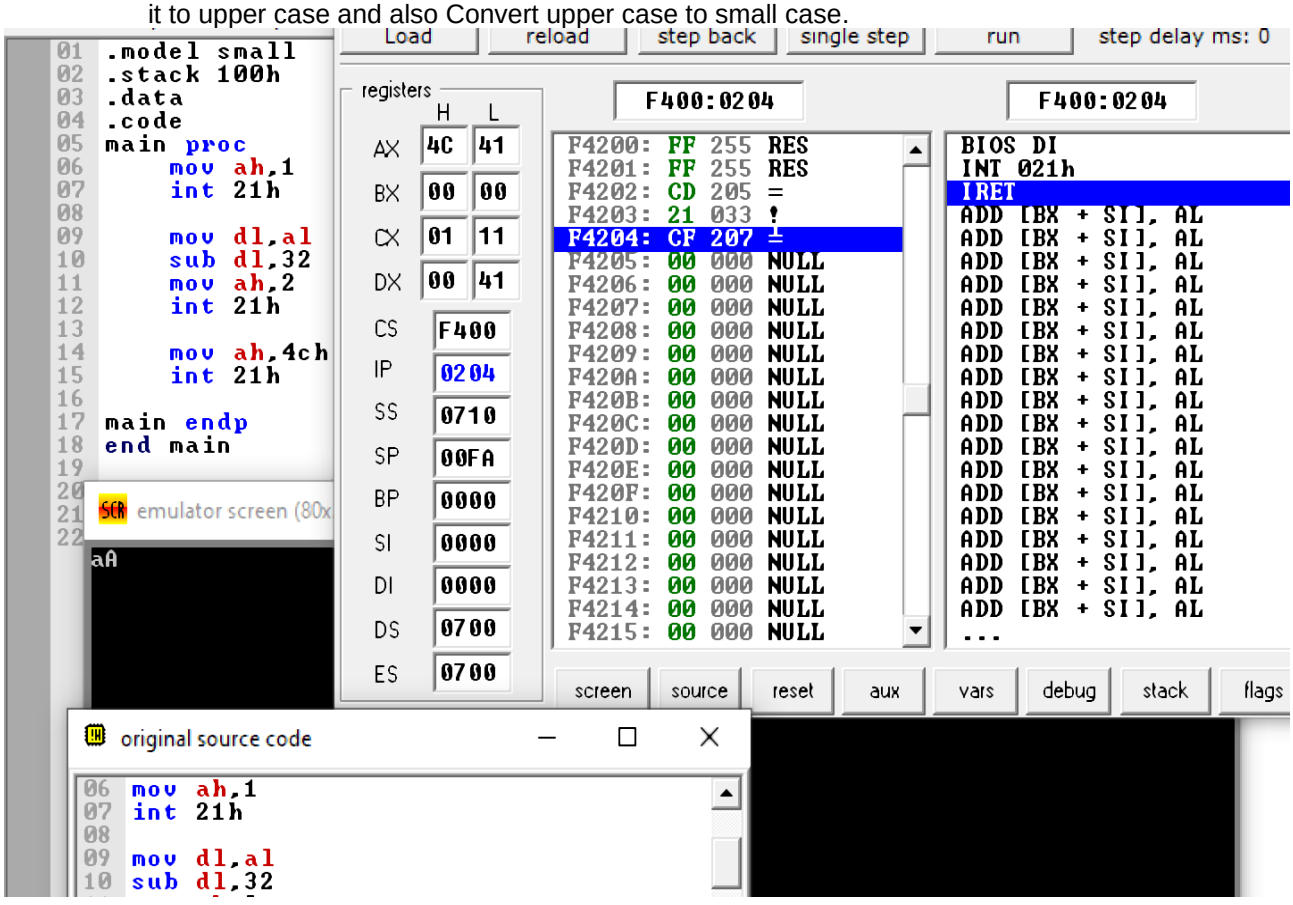
```
.model small
.stack 100h
.data

A db 'enter the String : $'

MESSAGE DB  0AH, 0DH,"your input were : $"

.code

main proc
     mov ax,@data
     mov ds,ax
     mov ah,9
     lea dx,a
     int 21h
       mov ah,1
       int 21h
         mov ah,1
       int 21h
         mov ah,1
       int 21h
         mov ah,1
       int 21h

      mov ax,@data
      mov ds,ax
      MOV DX, OFFSET MESSAGE
      mov ah,9
      int 21h

mov dl, al
add dl,61
sub dl,61
add dl,61
mov ah, 02h
int 21h
```

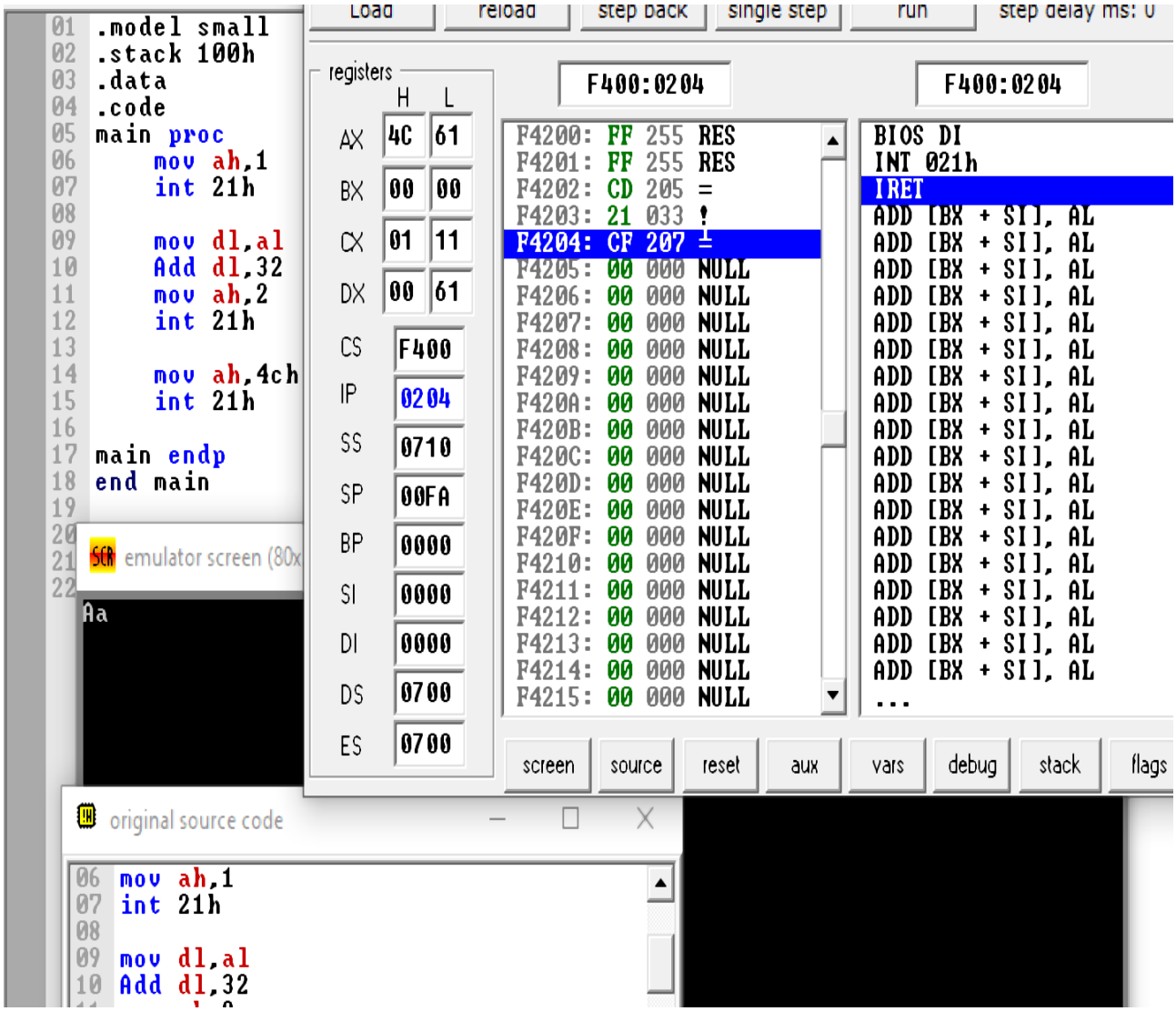
```
38  mov dl, al
39  mov ah, 02h
40  int 21h
41  mov dl, al
42  mov ah, 02h
43  int 21h
44  mov dl, al
45  mov ah, 02h
46  int 21h
47
48        mov ah,4ch
49        int 21h
50
51        main endp
52  end main
53
54
55
```






2- Write case conversion program, in which user enters small case character and program converts it to upper case and also Convert upper case to small case.

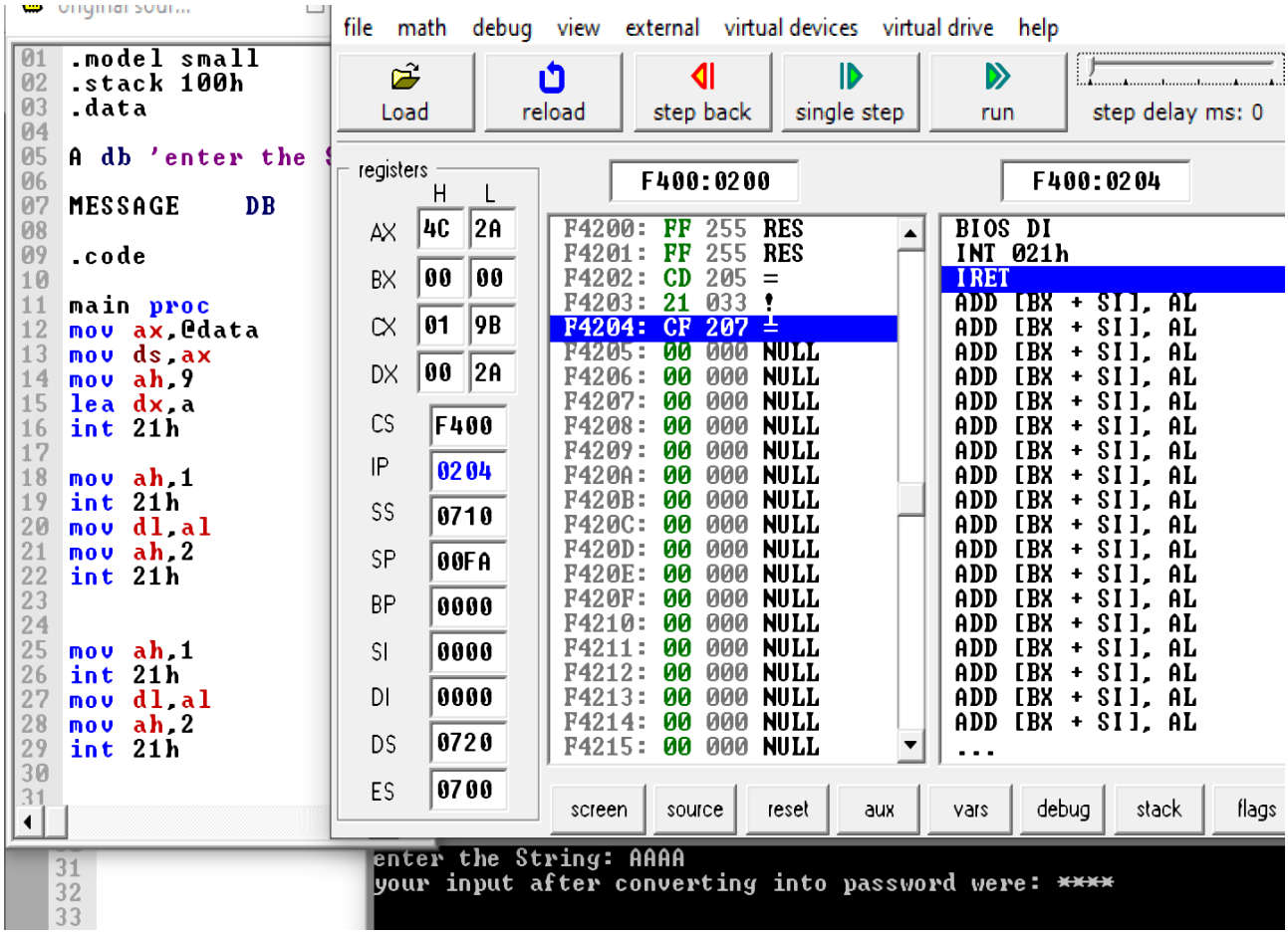





3- Write a program that print **** on the output screen.(4 character password)

```
.model small
.stack 100h
.data

A db 'enter the String : $'

MESSAGE DB  0AH, 0DH,"your input after converting into password were : $"

.code

main proc
    mov ax,@data
    mov ds,ax
    mov ah,9
    lea dx,a
    int 21h

      mov ah,1
      int 21h
      mov dl,al
      mov ah,2
      int 21h


        mov ah,1
      int 21h
      mov dl,al
      mov ah,2
      int 21h




     mov ax,@data
     mov ds,ax
     MOV DX, OFFSET MESSAGE
      mov ah,9
      int 21h

                  ;Displaying User's input
mov dl, al
add dl,6
mov ah, 02h
int 21h
mov dl, al
mov ah, 02h
int 21h
mov dl, al
mov ah, 02h
int 21h
mov dl, al
mov ah, 02h
int 21h

       mov ah,4ch
       int 21h

       main endp
end main
```

line: 10 col: 8 drag a file here to open


file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 4C 2A
BX 00 00
CX 01 9B
DX 00 2A
CS F400
IP 0204
SS 0710
SP 00FA
BP 0000
SI 0000
DI 0000
DS 0720
ES 0700
F400:0200
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 =
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
...
screen source reset aux vars debug stack flags
01 .model small
02 .stack 100h
03 .data
04
05 A db 'enter the
06
07 MESSAGE DB
08
09 .code
10
11 main proc
12 mov ax,@data
13 mov ds,ax
14 mov ah,9
15 lea dx,a
16 int 21h
17
18 mov ah,1
19 int 21h
20 mov dl,al
21 mov ah,2
22 int 21h
23
24
25 mov ah,1
26 int 21h
27 mov dl,al
28 mov ah,2
29 int 21h
30
31
32
33
enter the String: AAAA
your input after converting into password were: ****