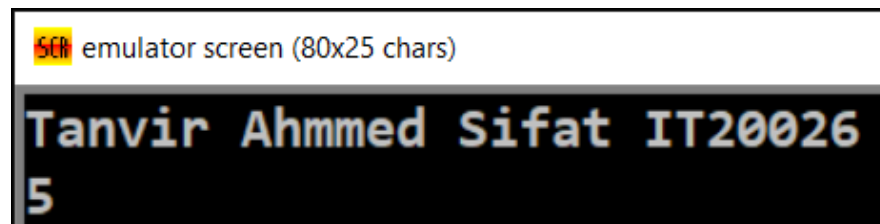


**Example 6.1: Suppose AX and BX contain signed numbers. Write some code to put the biggest one in CX.**

```
.model small                                mov ah,2
.stack 100h                                mov cx,bx
.data                                       mov dx,cx
msg1 db "Tanvir Ahmmed Sifat              add dx,48
IT20026$"                                int 21h
msg2 db 10,13,"$"                        jmp dos_exit
.code
L1:
main proc                                mov ah,2
    mov ax,@data                          mov cx,ax
    mov ds,ax                             mov dx,cx
    mov ah,9                              add dx,48
    lea dx,msg1                           int 21h
    int 21h                               dos_exit:
    lea dx,msg2
    int 21h
    mov ax,3                               mov ah,4ch
    mov bx,5                               int 21h
    cmp ax,bx
    jg L1
main endp
end main
```

**Output:**



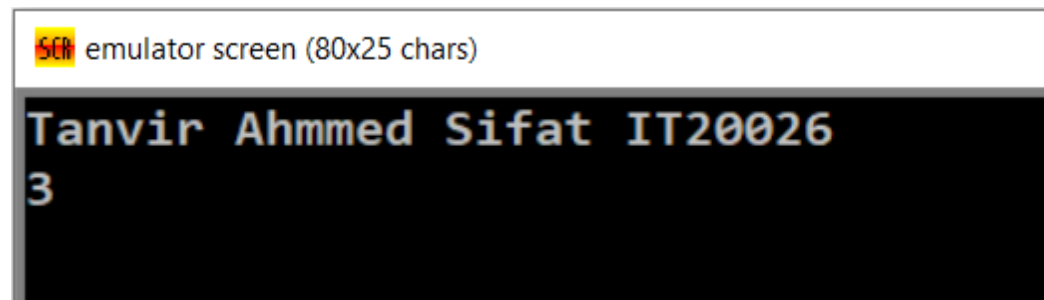
ScH emulator screen (80x25 chars)

```
Tanvir Ahmmed Sifat IT20026
5
```

## EX 6.2: REPLACE THE NUMBER IN AX BY ITS ABSOLUTE VALUE.

```
.model small                                mov ah,2
.stack 100h                                add ax,48
.data                                       mov dx,ax
msg1 db "Tanvir Ahmmed Sifat               int 21h
IT20026$"                                jmp exit
msg2 db 10,13,"$"
.code
main proc                                L1:
    mov ax,@data                            mov ah,2
    mov ds,ax                                add ax,48
    mov ah,9                                mov dx,ax
    lea dx,msg1                            int 21h
    int 21h                                exit:
    lea dx,msg2
    int 21h                                mov ah,4ch
    mov ax,-3                                int 21h
    cmp ax,0
    jge L1                                main endp
    neg ax                                end main
```

### Output:



```
SCM emulator screen (80x25 chars)
Tanvir Ahmmed Sifat IT20026
3
```

**Example 6.3: Suppose AL and BL contain extended ASCII characters. Display the one comes first in the character sequence.**

```
.model small                                cmp al,bl
.stack 100h                                jb L1
.data                                       mov ah,2
msg1 db "Tanvir Ahmmed Sifat               mov dl,bl
IT20026$"                                int 21h
msg2 db 10,13,"$"                        jmp dos_exit
.code                                       L1:
main proc                                mov ah,2
    mov ax,@data                          mov dl,al
    mov ds,ax                             int 21h
    mov ah,9
    lea dx,msg1                            dos_exit:
    int 21h
    lea dx,msg2                            mov ah,4ch
    int 21h                               int 21h

    mov al,83h                             main endp
    mov bl,85h                             end main
```

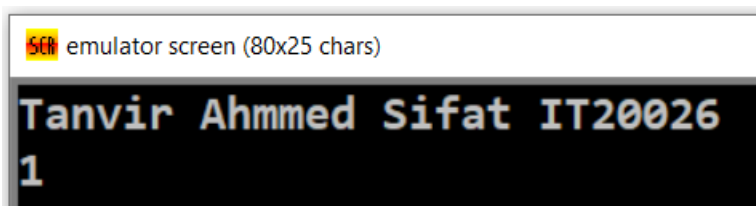
**Output:**

```
Scf emulator screen (80x25 chars)
Tanvir Ahmmed Sifat IT20026
f
```

**Example 6.4:** If AX contains a negative number , put - 1 in BX ; if AX contains 0, put 0 in BX ; if AX contains a positive number , put 1 in BX.

```
.model small                                jl negv
.stack 100h                                je zero
.data                                       jg pos
msg1 db "Tanvir Ahmmed Sifat               negv:
IT20026$"                                mov bx,-1
msg2 db 10,13,"$"                        jmp dos_exit
.code
main proc                                zero:
    mov ax,@data                        mov bx,0
    mov ds,ax                          jmp dos_exit
    mov ah,9                            pos:
    lea dx,msg1                        mov bx,1
    int 21h                            jmp dos_exit
    lea dx,msg2                        dos_exit:
    int 21h
    mov ah,1                            mov ah,4ch
    int 21h                            int 21h
    mov bx,ax
    cmp ax,0                            main endp
                                         end main
```

### **Output:**



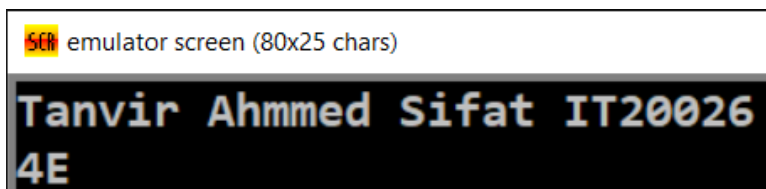
emulator screen (80x25 chars)

```
Tanvir Ahmmed Sifat IT20026
1
```

**Example 6.5: If AL contains 1 or 3 , display “O” ; If AL contains 2 or 4 , display “E”**

```
.model small                                je o
.stack 100h                                cmp al,'2'
.data                                       je e
msg1 db "Tanvir Ahmmed Sifat              cmp al,'4'
IT20026$"                                  je e
msg2 db 10,13,"$"                          e:
.code                                       mov ah,2
main proc                                  mov dl,'E'
    mov ax,@data                           int 21h
    mov ds,ax                               jmp dos_exit
    mov ah,9                               o:
    lea dx,msg1                             mov ah,2
    int 21h                                mov dl,'O'
    lea dx,msg2                             int 21h
    int 21h                                jmp dos_exit
    mov ah,1                               dos_exit:
    int 21h                                mov ah,4ch
    cmp al,'1'                             int 21h
    je o                                    main endp
    cmp al,'3'                             end main
```

**Output:**



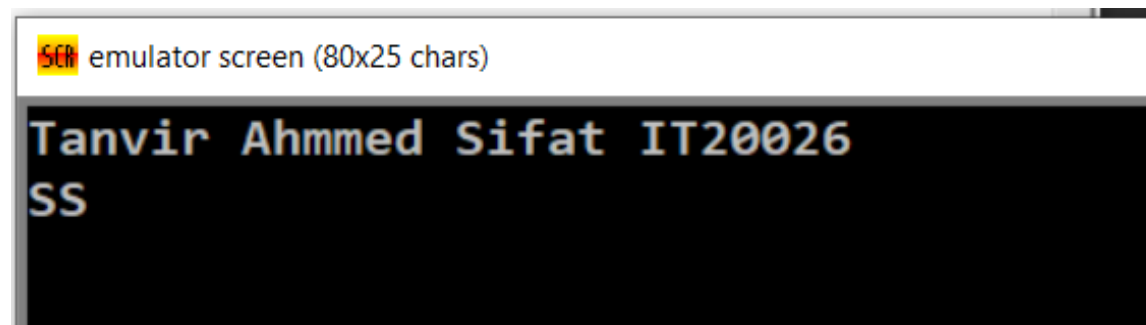
568 emulator screen (80x25 chars)

Tanvir Ahmmed Sifat IT20026  
4E

**Example 6.6: Read a character, and if it's an uppercase letter, display it.**

```
.model small                                int 21h
.stack 100h                                cmp al,'A'
.data                                       jl dos_exit
msg1 db "Tanvir Ahmmed Sifat               cmp al,'Z'
IT20026$"                                jg dos_exit
msg2 db 10,13,"$"                        mov ah,2
.code                                       mov dl,al
main proc                                int 21h
    mov ax,@data                          jmp dos_exit
    mov ds,ax                             dos_exit:
    mov ah,9
    lea dx,msg1                           mov ah,4ch
    int 21h                               int 21h
    lea dx,msg2
    int 21h
    mov ah,1                               main endp
                                           end main
```

**Output:**

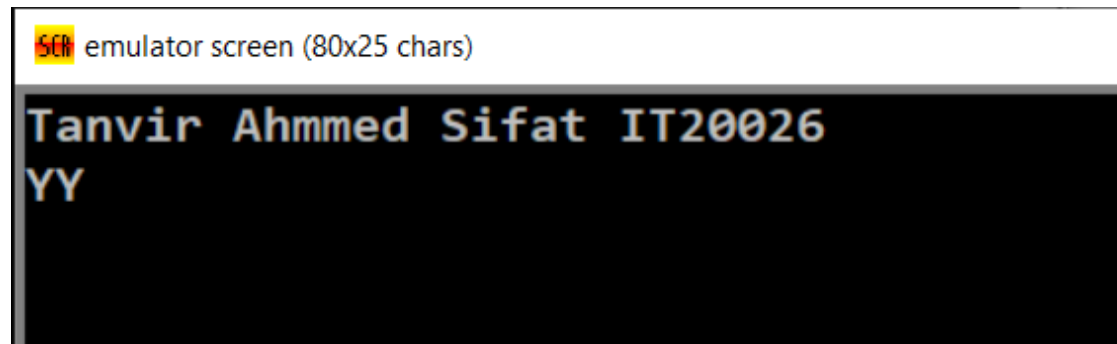


```
emulator screen (80x25 chars)
Tanvir Ahmmed Sifat IT20026
SS
```

**Example 6.7 : Read a character if it's "y" or "Y" , display it ; otherwise, terminate the program.**

```
.model small                                je show
.stack 100h                                cmp al,'Y'
.data                                       je show
msg1 db "Tanvir Ahmmed Sifat              jmp dos_exit
IT20026$"
msg2 db 10,13,"$"
.code
main proc
    mov ax,@data
    mov ds,ax
    mov ah,9
    lea dx,msg1
    int 21h
    lea dx,msg2
    int 21h
    mov ah,1
    int 21h
    cmp al,'y'
    je show
    mov ah,2
    mov dl,al
    int 21h
    jmp dos_exit
dos_exit:
    mov ah,4ch
    int 21h
main endp
end main
```

**Output:**

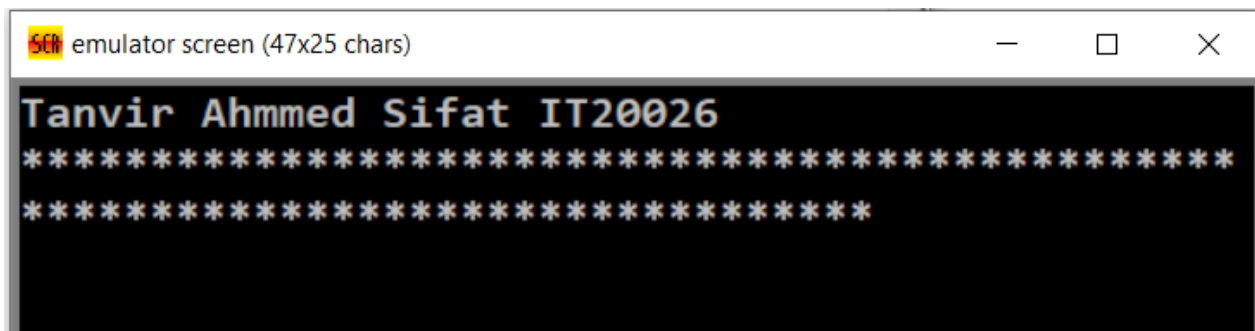


```
SCR emulator screen (80x25 chars)
Tanvir Ahmmed Sifat IT20026
YY
```

### EX 6.8: WRITE A COUNT -CONTROLLED LOOP TO DISPLAY A ROW OF 80 STARS.

```
.model small                                mov cx,80
.stack 100h                                jcxz exit
.data                                       L:
msg1 db "Tanvir Ahmmed Sifat               mov ah,2
IT20026$"                                mov dx,"*"
msg2 db 10,13,"$"                        int 21h
.code                                       loop l
main proc
    mov ax,@data
    mov ds,ax
    mov ah,9
    lea dx,msg1
    int 21h
    lea dx,msg2
    int 21h
    mov ah,4ch
    int 21h
main endp
end main
exit:
```

### Output:



```
ScH emulator screen (47x25 chars)
Tanvir Ahmmed Sifat IT20026
*****
*****
```




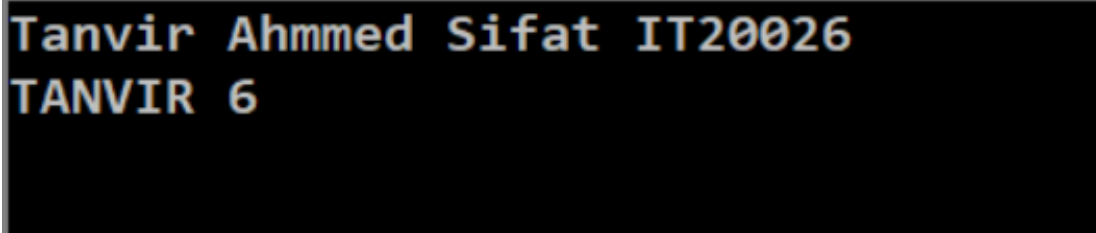
**EX 6.9: Write some code to count the number of characters in an input line**

```
.model small                                int 21h
.stack 100h                                inc bl
.data                                       cmp al,20h
msg1 db "Tanvir Ahmmed Sifat              je show
IT20026$"                                jmp do
msg2 db 10,13,"$"                        show:
.code                                       mov ah,2
main proc                                sub bl,1
    mov ax,@data                          mov dl,bl
    mov ds,ax                             int 21h
    mov ah,9                              exit:
    lea dx,msg1
    int 21h                                mov ah,4ch
    lea dx,msg2                            int 21h
    int 21h
    mov bl,30h
    do:                                    main endp
    mov ah,1                              end main
```

**Output:**

---

 emulator screen (80x25 chars)




```
Tanvir Ahmmed Sifat IT20026
TANVIR 6
```

**Example 6.10 : Write some code to read characters until a blank is read.**

```
.model small                                int 21h
.stack 100h                                inc bl
.data                                       cmp al,20h
msg1 db "Tanvir Ahmmed Sifat              je exit
IT20026$"                                jmp do
msg2 db 10,13,"$"                        show:
.code                                       mov ah,2
main proc                                sub bl,1
    mov ax,@data                          mov dl,bl
    mov ds,ax                             int 21h
    mov ah,9                              exit:
    lea dx,msg1
    int 21h                                mov ah,4ch
    lea dx,msg2                            int 21h
    int 21h
    mov bl,30h
                                           main endp
do:                                       end main
    mov ah,1
```

**Output:**

 emulator screen (80x25 chars)

