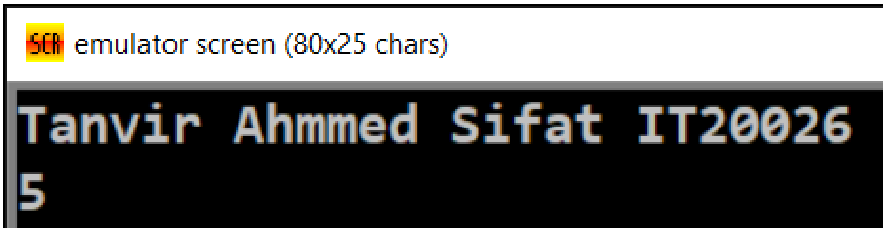
Example 6.1: Suppose AX and BX contain signed numbers. Write some code to put the biggest one in CX.

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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 ax,3
 mov bx,5
 cmp ax,bx
 jg L1
mov ah,2
mov cx,bx
mov dx,cx
add dx,48
int 21h
jmp dos_exit
L1:
mov ah,2
mov cx,ax
mov dx,cx
add dx,48
int 21h
dos_exit:

mov ah,4ch
int 21h


main endp
end main
```

<u>Output:</u>

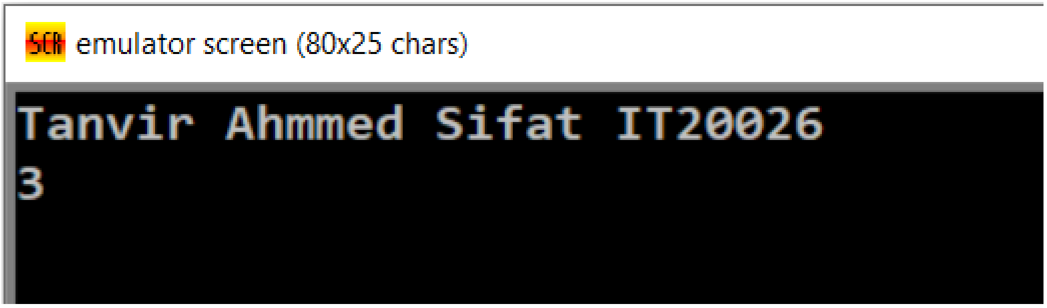


```
Tanvir Ahmmed Sifat IT20026
5
```

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

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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 ax,-3
 cmp ax,0
 jge L1
 neg ax
mov ah,2
add ax,48
mov dx,ax
int 21h
jmp exit

L1:
mov ah,2
add ax,48
mov dx,ax
int 21h

exit:

mov ah,4ch
int 21h

main endp
end main
```

Output:

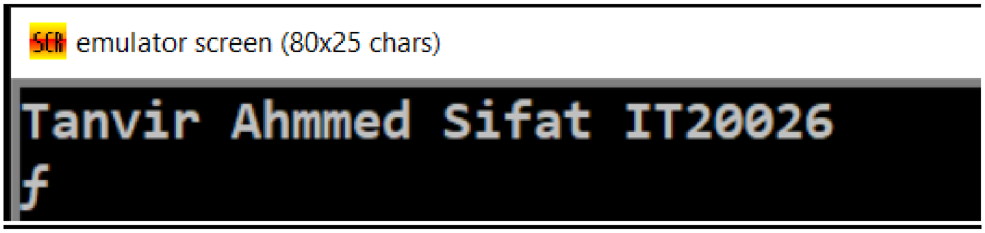


```
Tanvir Ahmmed Sifat IT20026
3
```

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

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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 al,83h
 mov bl,85h
cmp al,bl
jb L1
mov ah,2
mov dl,bl
int 21h
jmp dos_exit
L1:
mov ah,2
mov dl,al
int 21h

dos_exit:

mov ah,4ch
int 21h

main endp
end main
```

Output:

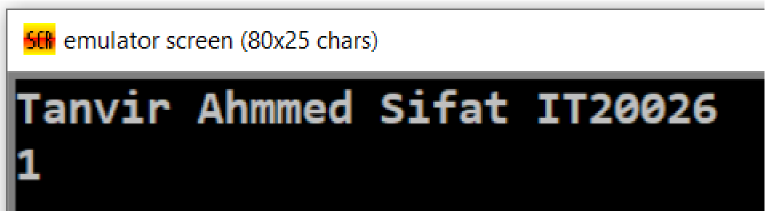


```
Tanvir Ahmmed Sifat IT20026
ƒ
```

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
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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
 mov bx,ax
 cmp ax,0
jl negv
je zero
jg pos
negv:
mov bx,-1
jmp dos_exit
zero:
mov bx,0
jmp dos_exit
pos:
mov bx,1
jmp dos_exit
dos_exit:

mov ah,4ch
int 21h

main endp
end main
```

Output:

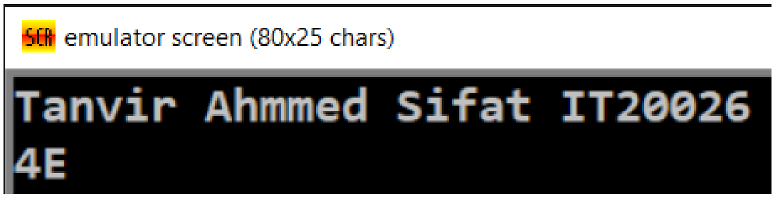


```
Tanvir Ahmmed Sifat IT20026
1
```

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

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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,'1'
 je o
  cmp al,'3'
je o
 cmp al,'2'
je e
 cmp al,'4'
je e
e:
mov ah,2
mov dl,'E'
int 21h
jmp dos_exit
o:
mov ah,2
mov dl,'O'
int 21h
jmp dos_exit
dos_exit:
mov ah,4ch
int 21h
main endp
end main
```

Output:

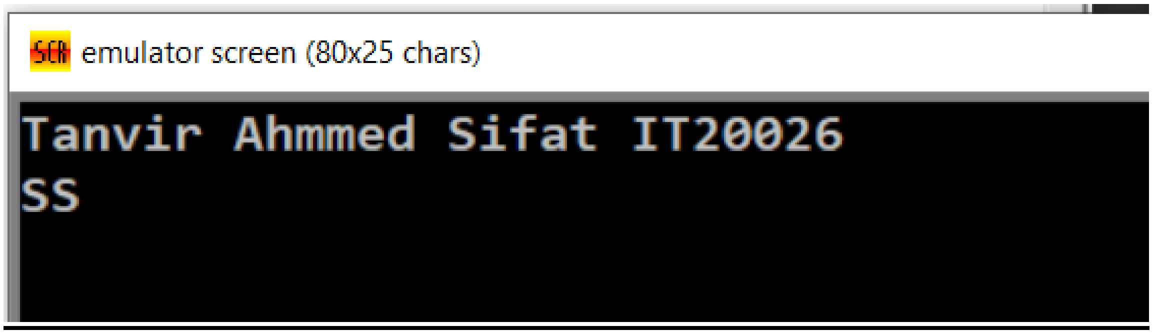


```
Tanvir Ahmmed Sifat IT20026
4E
```

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

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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,'A'
 jl dos_exit
 cmp al,'Z'
 jg dos_exit
 mov ah,2
 mov dl,al
 int 21h
 jmp dos_exit
 dos_exit:

 mov ah,4ch
 int 21h

 main endp
end main
```

Output:

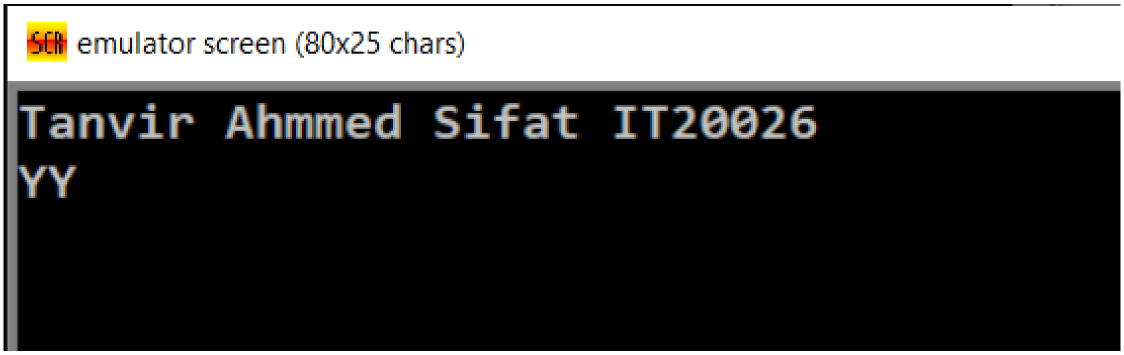


```
Tanvir Ahmmed Sifat IT20026
SS
```

Example 6.7 : Read a character it it’s “y” or “Y” , display it ; otherwise, terminate the program.

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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
 cmp al,'Y'
 je show
 jmp dos_exit

 show:
 mov ah,2
 mov dl,al
 int 21h
 jmp dos_exit
 dos_exit:

 mov ah,4ch
 int 21h

 main endp
end main
```

Output:

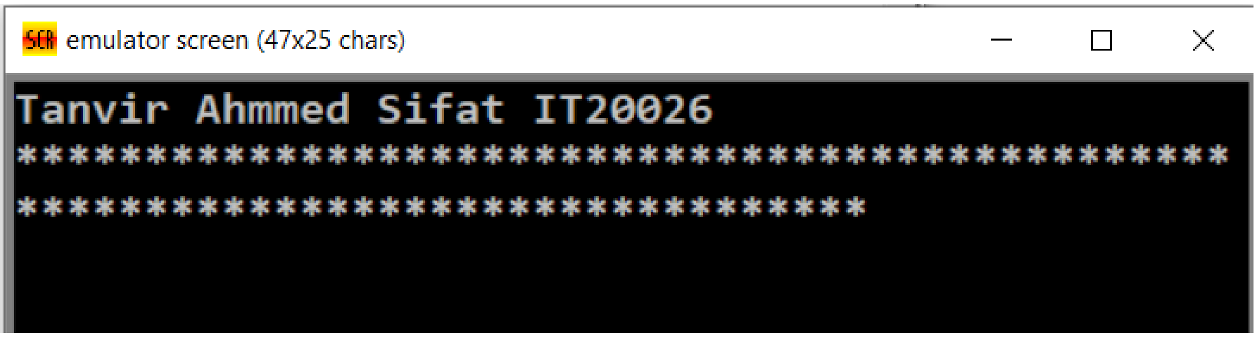


```
Tanvir Ahmmed Sifat IT20026
YY
```

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

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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 cx,80
 jcxz exit
 L:
 mov ah,2
 mov dx,"*"
 int 21h
 loop l

 exit:

 mov ah,4ch
 int 21h

 main endp
end main
```

Output:

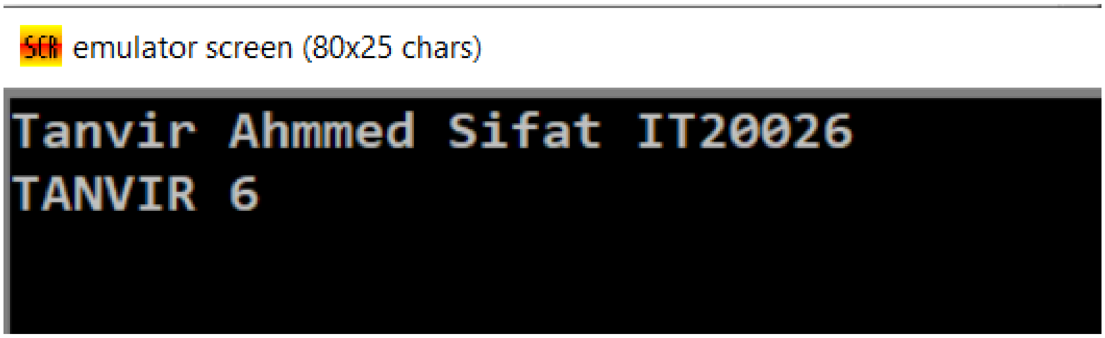


```
Tanvir Ahmmed Sifat IT20026
***********************************************
*********************************
```

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

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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 bl,30h
 do:
 mov ah,1
int 21h
inc bl
cmp al,20h
je show
jmp do
show:
mov ah,2
sub bl,1
mov dl,bl
int 21h
exit:

mov ah,4ch
int 21h

main endp
end main
```

Output:



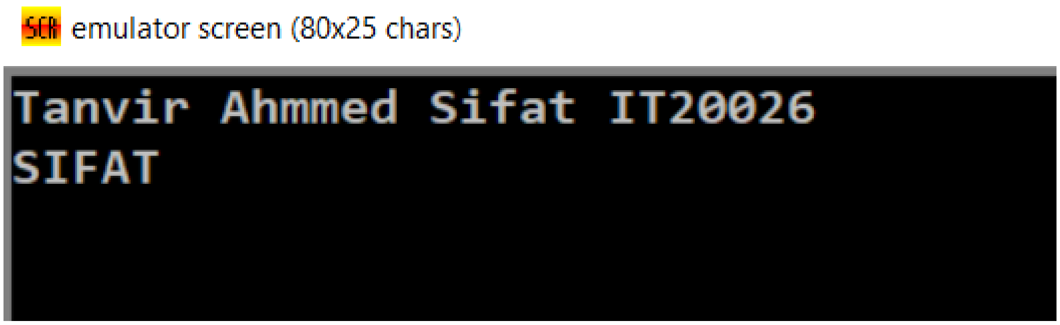
```
Tanvir Ahmmed Sifat IT20026
TANVIR 6
```

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

```
.model small
.stack 100h
.data
msg1 db "Tanvir Ahmmed Sifat 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 bl,30h
 do:
 mov ah,1
```

```
 int 21h
 inc bl
 cmp al,20h
 je exit
 jmp do
 show:
 mov ah,2
 sub bl,1
 mov dl,bl
 int 21h
 exit:

 mov ah,4ch
 int 21h

 main endp
end main
```

<u>Output:</u>



emulator screen (80x25 chars)

```
Tanvir Ahmmed Sifat IT20026
SIFAT
```