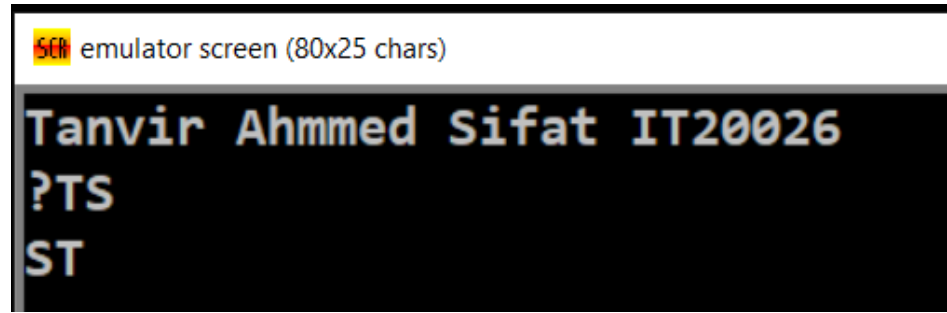


**Exercise 8:**Write a program to display a “?” , read two capital letters, and display them on the next line in alphabetical order.

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

## Output:



```
SCSI emulator screen (80x25 chars)
Tanvir Ahmmed Sifat IT20026
?TS
ST
```

**Exercise 9:** Write a program to display the extended ASCII characters. Display 10 characters per line, separated by blanks. Stop after the extended character have been displayed once.

## Solution:

```
.model small                mov cx,127
.stack 100h                 mov bl,0
.data                       show:
msg1 db "Tanvir Ahmmed Sifat  mov ah,2
IT20026$"                  inc cx
msg2 db 10,13,"$"          cmp cx,255
.code                       jg dos_exit
main proc                  mov dx,cx
mov ax,@data               int 21h
mov ds,ax                  mov dx,32
mov ah,9                   int 21h
lea dx,msg1                jmp go
int 21h                     go:
lea dx,msg2                inc bl
int 21h                     cmp bl,10
```

```

je print
jmp show
print:
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

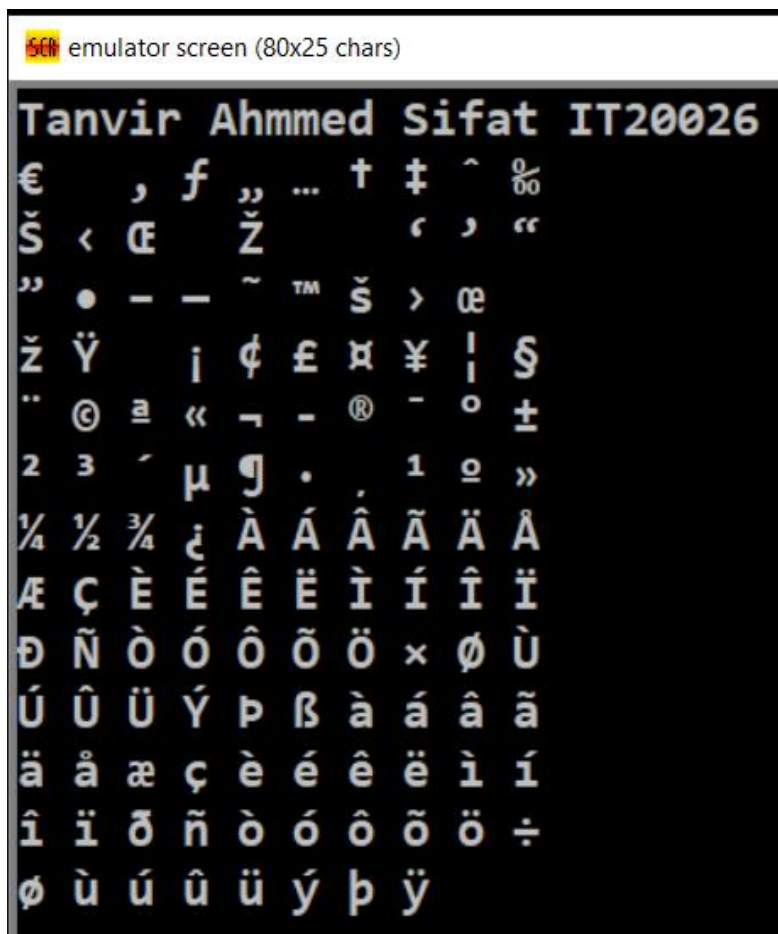
```

```

mov bl,0
jmp show
dos_exit:
mov ah,4ch
int 21h
main endp
end main

```

### Output:



**Exercise - 10:** Write a program that will prompt the user to enter a hex digit character ("0" ...."9" or "A" .."F") display it on the next line in decimal, and ask the user if he or she wants to do it again. If the user types anything else, the program terminates. If the user enters an illegal character, prompt the user to try again.

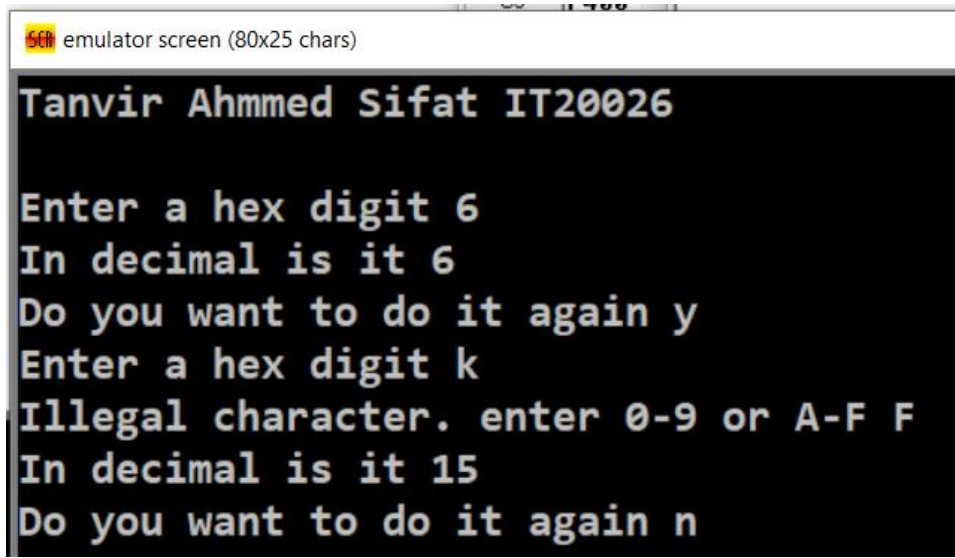
**Solution:**

.model small	again:
.stack 100h	mov ah,9
.data	lea dx,mg1
a db 'Tanvir Ahmmed Sifat IT20026\$'	int 21h
b db 10,13,'\$'	mov ah,1
mg1 db 10,13,'Enter a hex digit \$'	int 21h
mg2 db 10,13,'In decimal is it \$'	mov bl,al
mg3 db 10,13,'Do you want to do it again \$'	jmp do
mg4 db 10,13,'Illegal character. enter 0-9 or A-F \$'	do:
.code	cmp bl,'9'
main proc	jg hex
mov ax,@data	je num
mov ds,ax	j1 num
mov ah,9	hex:
lea dx,a	cmp bl,'F'
int 21h	jg illegal
lea dx,b	mov ah,9
int 21h	lea dx,mg2
	int 21h
	mov dl,49

```
mov ah,2
int 21h
mov ah,2
sub bl,17
mov dl,bl
int 21h
jmp inp
inp:
mov ah,9
lea dx,mg3
int 21h
mov ah,1
int 21h
mov cl,al
cmp cl,'Y'
je again
cmp cl,'y'
je again
jmp exit
num:
cmp bl,'0'
jl illegal
```

```
lea dx,mg2
int 21h
mov ah,9
int 21h
mov dl,bl
mov ah,2
int 21h
jmp inp
illegal:
lea dx,mg4
mov ah,9
int 21h
mov ah,1
int 21h
mov bl,al
jmp do
exit:
mov ah,4ch
int 21h
main endp
end main
```

## Output:



```
emulator screen (80x25 chars)
Tanvir Ahmmed Sifat IT20026
Enter a hex digit 6
In decimal is it 6
Do you want to do it again y
Enter a hex digit k
Illegal character. enter 0-9 or A-F F
In decimal is it 15
Do you want to do it again n
```

**Exercise 11:** Do programming exercise 10, except that if the user fails to enter a hex-digit character in three tries, display a message and terminate the program.

## Solution:

<code>.model small</code>	<code>mge db 10,13,'Too Many Try\$'</code>
<code>.stack 100h</code>	<code>.code</code>
<code>.data</code>	<code>main proc</code>
<code>a db 'Tanvir Ahmmed Sifat</code> <code>IT20026\$'</code>	<code>mov ax,@data</code>
<code>b db 10,13,'\$'</code>	<code>mov ds,ax</code>
<code>mga db 10,13,'ENTER A HEX</code> <code>DIGIT:\$'</code>	<code>mov ah,9</code>
<code>mgb db 10,13,'IN DECIMAL IS IT:\$'</code>	<code>lea dx,a</code>
<code>mgc db 10,13,'DO YOU WANT TO</code> <code>DO IT AGAIN? \$'</code>	<code>int 21h</code>
<code>mgd db 10,13,'ILLEGAL</code> <code>CHARACTER- ENTER 0-9 OR A-</code> <code>F:\$'</code>	<code>lea dx,b</code>
	<code>int 21h</code>
	<code>again:</code>
	<code>mov cx,0</code>

```
lea dx,mga
mov ah,9
int 21h
mov ah,1
int 21h
mov bl,al
jmp go
go:
cmp bl,'9'
jg hex
jl num
je num
hex:
cmp bl,'F'
jg illegal
lea dx,mgb
mov ah,9
int 21h
mov dl,49d
mov ah,2
int 21h
sub bl,17d
mov dl,bl
mov ah,2
int 21h
```

```
jmp inp
inp:
lea dx,mgc
mov ah,9
int 21h
mov ah,1
int 21h
mov cl,al
cmp cl,'y'
je again
cmp cl,'Y'
je again
jmp exit
num:
cmp bl,'0'
jl illegal
lea dx,mgb
mov ah,9
int 21h
mov dl,bl
mov ah,2
int 21h
jmp inp
illegal:
inc cx
```

cmp cx,3	lea dx,mge
je i2	mov ah,9
lea dx,mgd	int 21h
mov ah,9	jmp exit
int 21h	exit:
mov ah,1	mov ah,4ch
int 21h	int 21h
mov bl,al	main endp
jmp go	end main
i2:	

### **Output:**

```
568 emulator screen (80x25 chars)
Tanvir Ahmmed Sifat IT20026
ENTER A HEX DIGIT:F
IN DECIMAL IS IT:15
DO YOU WANT TO DO IT AGAIN?y
ENTER A HEX DIGIT:x
ILLEGAL CHARACTER- ENTER 0-9 OR A-F:y
ILLEGAL CHARACTER- ENTER 0-9 OR A-F:z
Too Many Try
```



**Exercise 12:** Write a program that reads a string of capital letters, ending with a carriage return, and displays the longest sequence of consecutive alphabetically increasing capital letters read.

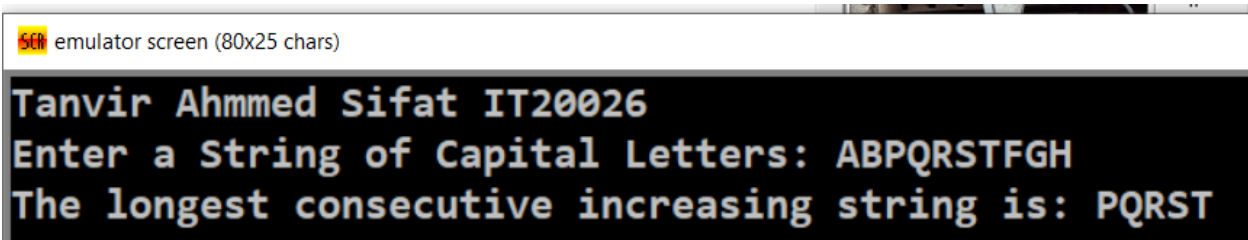
**Solution:**

.MODEL SMALL	INPUT_1:
.STACK 100H	MOV AH,1
.DATA	INT 21H
x DB 'Tanvir Ahmmed Sifat IT20026\$'	CMP AL,0DH
y DB 10,13,'\$'	JE END_
inmsg DB 'Enter a String of Capital Letters : \$'	MOV CL,1
outmsg DB 0DH,0AH,'The longest consecutive increasing string is : \$'	MOV BL,AL
.CODE	MOV DH,AL
MAIN PROC	INPUT_2:
MOV AX,@DATA	INT 21H
MOV DS,AX	CMP AL,0DH
MOV AH,9	JE END_
LEA DX,x	INC BL
INT 21H	CMP BL,AL
LEA DX,y	JNE INIT
INT 21H	INC CL
MOV CH,0	JMP INPUT_2
MOV AH,9	INIT:
LEA DX,inmsg	CMP CH,CL
INT 21H	JL UPDATE
	MOV CL,1
	MOV BL,AL

```
MOV DH,AL
JMP INPUT_2
UPDATE:
MOV CH,CL
MOV BH,DH
MOV CL,1
MOV BL,AL
MOV DH,AL
JMP INPUT_2
END_:
CMP CH,CL
JL RUPDATE
JMP END_2
RUPDATE:
MOV CH,CL
MOV BH,DH
JMP END_2
END_2:
```

```
MOV AH,9
LEA DX,outmsg
INT 21H
MOV AH,2
MOV DL,BH
OUTPUT:
CMP CH,0
JE FINISH
DEC CH
INT 21H
INC DL
JMP OUTPUT
FINISH:
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

### **Output:**



emulator screen (80x25 chars)

```
Tanvir Ahmmed Sifat IT20026
Enter a String of Capital Letters: ABPQRSTFGH
The longest consecutive increasing string is: PQRST
```