Case Conversion

Expt No:	8	**Name:**	Shivanirudh S G
Date :	16/10/2020	**Reg No:**	185001146

Aim:

To convert case of characters from keyboard on the move in 8086.

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move value of count to CX register.
- Move value 1 to AH register, under label L1.
- Request interrupt 21H to enable keyboard input stored AL register.
- Compare AL with hex value 60 using CMP AL, 60H.
- Jump to label UPPER if AL is greater than 60.
- Add hex value 20H to AL using ADD AL, 20H.
- Jump to label DONE.
- Subtract hex value 20H from AL using SUB AL, 20H under label UPPER.
- Move value 2 to AH register, under label DONE.
- Move contents to AL to DL register to display it.
- Request interrupt 21H to enable output at standard output.
- Loop back to label L1.

Program:

Program	**Comments**
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
count equ 10H	Define variable count with value 10H
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov cx, count	Move value of count to CX register
l1: mov ah, 1	Move value 1 o AH register
int 21h	Accept keyboard input
cmp al, 60H	Compare value in AL register with hex value 60
jnc upper	Jump to upper if AL ¿ 60
add al, 20H	AL = AL + 20
jmp done	Jump to done
upper: sub al, 20H	AL = AL - 20
done: mov ah, 2	Move value 2 to AH register
mov dl, al	Move value in AL register to DL register
int 21h	Display value in standard output
loop l1	Loop back from L1
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:

```
-u
0E24:0000 B8240E        MOV     AX,0E24
0E24:0003 8ED8          MOV     DS,AX
0E24:0005 B91000        MOV     CX,0010
0E24:0008 B401          MOV     AH,01
0E24:000A CD21          INT     21
0E24:000C 3C60          CMP     AL,60
0E24:000E 7304          JNB     0014
0E24:0010 0420          ADD     AL,20
0E24:0012 EB02          JMP     0016
0E24:0014 2C20          SUB     AL,20
0E24:0016 B402          MOV     AH,02
0E24:0018 8AD0          MOV     DL,AL
0E24:001A CD21          INT     21
0E24:001C E2EA          LOOP    0008
0E24:001E B44C          MOV     AH,4C
-
```

Input and Output:

```
-g
Aa
-zZ
-Bb
-yY
-Cc
-xX
-Dd
-wW
-
D:\>_
```

Figure 1: **Input:** A, z, B, y, C, x, D, w ;
Output: a, Z, b, Y, c, X, d, W

Result:

The 8086 programs were written to perform matrix operations, and the results observed.