End

PROGRAM

```
assume cs:code, ds:data
data segment
list db 02h,04h,01h,09h,10h,07h
res db 01h dup(0)
data ends
code segment
start: mov ax,data
mov ds, ax
mov si, offset list
```



```
mov cl, 06h
mov al, 01h
up: cmp al, [si]
je down
inc si
dec cl
jnz up
mov res, 'N'
jmp l1
down: mov res, 'Y'
l1: int 03h
code ends
end start
```

Yes

Stop

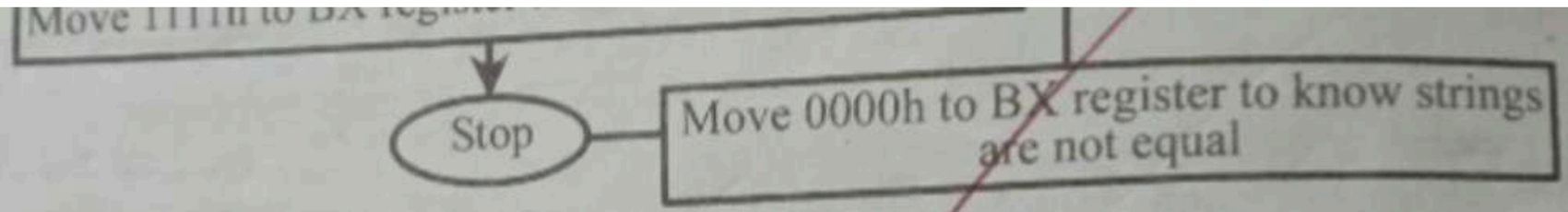
PROGRAM

```
assume cs:code , ds:data , es: extra
 data segment
 x1 db 'processor'
 data ends
 extra segment
 x2 db 09h dup (0)
 extra ends
 Code segment
Start: mov ax, data
 mov ds, ax.
```



```
mov ax, extra
mov es, ax
mov si, offset x1
mov di, offset x2
mov cl, 09h
cld
rep movsb
int 03h
code ends
end start.
```

LIST FILE	COMMENTS

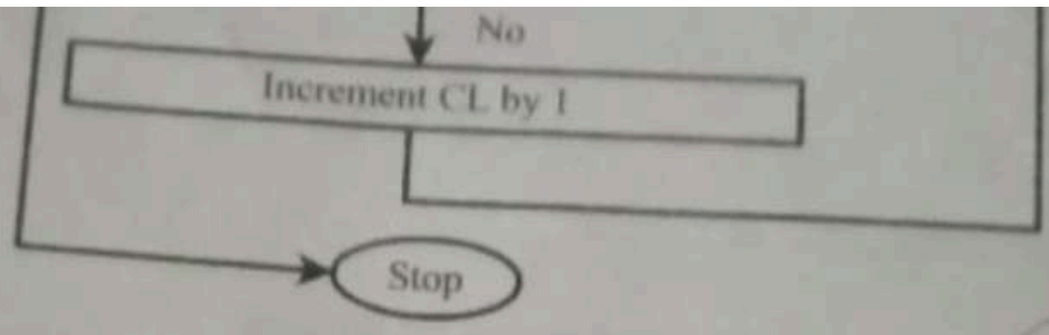


PROGRAM

```
assume cs:code, ds:data, es:extra
data segment
s1 db 'micro'
res db 05h dup(0)
data ends
```



```
extra segment
s2 db 'micro'
extra ends
code segment
start: mov ax, data
mov ds, ax
mov ax, extra
mov es, ax
mov si, offset s1
mov di, offset s2
mov cl, 09h
cld
l2: cmp sb
jne l1
loop l2
mov res, 'Y'
jmp l3
l1: mov res, 'n'
l3: int 03h
code ends
end start
```

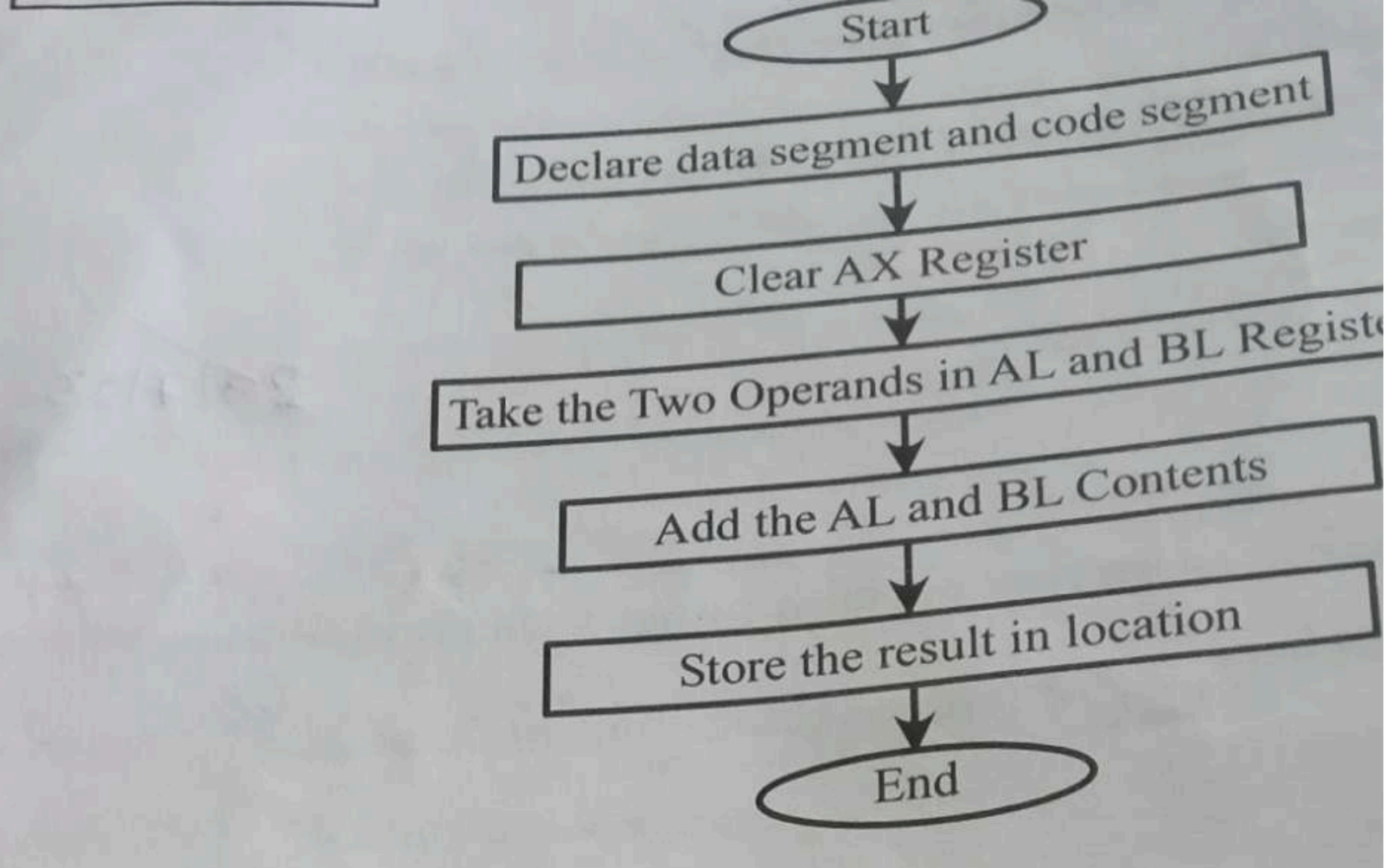


PROGRAM

```
assume cs:code, es:extra
extra segment
si db 'micro$'
extra ends
code segment
start: mov ax, data
mov es, ax
mov di, offset si
mov al, '$'
mov cl, 0ch
cld
l2 : scasb
jc l1
inc cl
jmp l2
l1: int 03h
code ends
end start
```

LIST FILE

ADDRESS	OPCODE	OPERATIONS	COMMENTS
			moving data in



PROGRAM

```
assume  cs:code , ds:data
data segment
a1 db  25h
a2  db  35h
data ends
Code segment
Start: mov  ax, data
mov  ds,ax
mov  al,a1
mov  bl,a2
add  al,bl
int 03h
Code ends
end start
```

```
assume cs:code,ds:data
data segment
list db 02h,07h,06h,05h,04h,08h
Count equ 06h
data ends
Code segment
start:mov ax,data
mov ds,ax
mov dl,Count-1
back:mov cl,dl
mov si;offset list
again:mov al,[si]
cmp al,[si+1]
jb go
xchg[si+1],al
xchg[si],al
go:inc si
loop again
dec dl
jnz back
int 03h
Code ends
end start
```

PROGRAM

```
assume cs:code,ds:data
data segment
list db 08h,06h,07h,01h,02h,03h
Count equ 06h
data ends
Code segment
Start:mov ax,data
mov ds,ax
mov dl,Count-1
back :mov cl,dl
mov si,offset list
again :mov al,[si]
cmp.al,[si+1]
ja go
xchg [si+1],al
xchg [si],al
go:inc si
loop again
dec dl
jNz back
int 03h
Code ends
end start.
```

OBJECTIVE

To write an assembly language program to display system time

TOOLS REQUIRED

PC installed with MASM

PROGRAM

```
assume CS:code
Code segment
Start:
        mov ah, 2ch
        int 21h
        mov al, ch
        aam
        mov bt, ax
        Call display
        mov dl, ':'
        mov ah, 02h
        int 21h
        mov al, cl
        aam
        mov bt, ax
        Call display
        mov dl, ':'
        mov ah, 02h
        int 21h
        mov ah, 2ch
        int 21h
```

```
        mov al,dh
        aam
        mov bx,ax
        call display
        mov ah,4ch
        int 21h
display proc
        mov dl,bh
        add dl,30h
        mov ah,02h
        int 21h
        mov dl,bl
        add dl,30h
        mov ah,02h
        int 21h
        ret
display endp
Code ends
end start.
```

o/p verified

OBJECTIVE

To write an assembly language program to display system time

TOOLS REQUIRED

PC installed with MASM

PROGRAM

```
assume cs:code
Code segment
start:
        mov ah,2ch
        int 21h
        mov al,ch
        aam
        mov bx,ax
        Call display
        mov dl,':'
        mov ah,02h
        int 21h
        mov al,cl
        aam
        mov bx,ax
        Call display
        mov dl,':'
        mov ah,02h
        int 21h
        mov ah,2ch
        int 21h
```



```
        mov al,dh
        aam
        mov bx,ax
        Call display
        mov ah,4ch
        int 21h
display proc
        mov dl,bh
        add dl,30h
        mov ah,02h
        int 21h
        mov dl,bl
        add dl,30h
        mov ah,02h
        int 21h
        ret
display endp
Code ends
end start.
```

o/p verified

LIST FILE

9. Go to file → Download Hex file. Select the HEX file by following the path E drive → Talk → MC COMM → Nifc26 → Nifc26. Now I appear on TALK window to indicate that file is downloaded.

10. Type G E000 (Starting address) and press enter.

11. Now program gets executed

12. Now data location displays on the LCD display of Kit and by pressing SW1 of study card we can observe the increment in the data field displayed.

PROGRAM TO VERIFY TIMER '0'- COUNTER MODE

ADDRESS	OPCODE	LABEL	OPERATIONS

			MOV A, TMOD
			ORL A, #05H
			MOV TMOD, A
			SETB TR0
			LOOP LCALL 03EAH
			MOV DPTR, #0191H
			MOV A, TL0
			MOVX @DPTR, A
			INC DPTR
			MOV A, TH0
			MOVX @DPTR, A
			LCALL 0748H
			SJMP LOOP

Execution

1) Short jp1 of 1&2 pins and press sw1 for manual increment

2) Short jp1 of 2&3 pins for auto increment

PROGRAM TO VERIFY TIMER-1 COUNTER MODE

ADDRESS	OPCODE	LABEL	OPERATIONS
			MOV A, TMOD
			ORL A, #50h
			MOV TMOD, A
			SETB TR1
		Loop	LCALL 6&EAH
			MOV DPTR, #019H
			MOV A, TL1
			MOVX @DPTR, A
			INC DPTR
			MOV A, TH1
			MOV @DPTR, A
			LCALL 6718H
			SJMP loop

Execution

1) Short jp1 of 5&6 pins and press sw2 for manual increment
2) Short jp2 of 4&5 pins for auto increment

PROGRAM TO STUDY TIMER-1 GATED MODE

ADDRESS	OPCODE	LABEL	OPERATIONS

h. G-4000(on system keyboard), we can observe the output on 8251 kit.

i. Remove RS232 cable from 8051 kit and connect it to 8251, transmitted data displayed on PC Monitor

PROGRAM

TRANSMISSION

I
```
ALS 8051/31 monitor V1.0
8051>A
Enter START address = 8000
8000   MOV A,#03
8000   7403   MOV A,#03
8002   MOV B,#05
8002   750B05   MOV 0B,#05
8005   ADD A,B
8005   250B   ADD A,B
8007   LCALL 03
8007   120003   LCALL 03
800A
8051>
ALS 8051/31 monitor V1.0
```

II
```
8051>G
PROGRAMME EXECUTION---
Enter START address = 8000
ACC  PSW  BPH  DPL  PCH  PCL  SP  B
07   01   00   00   80   0A   60  00
R0   R1   R2   R3   R4   R5   R6  R7
00   00   00   00   00   00   00  00
```


RECEPTION

II
```
ALS 8051/31 monitor V1.0
8051>G
PROGRAMME EXECUTION---
Enter START address = 8000
 ACC  PSW  DPH  DPL  PCH  PCL  SP  B
 07   01   00   00   80   0A   60  00
 R0   R1   R2   R3   R4   R5   R6  R7
 00   00   00   00   00   00   00  00
```

RESULT

Hence, established Communication b/w 8051 and PC by writing program using ALP.

8. G-4000(on system keyboard), we can observe the output on 8251 kit
9. Remove RS232 cable from 8086kit and connect it to 8251, transmitted data displayed on PC Monitor

RECEPTION

1. Connect 8086 kit PC using RS232 cable.
2. Connect Power supply to 8086 kit and 8251 interfacing kit (only blue(+5V) and black(0V) lines Power cable to power supply)
3. Connect 8251 to 8086 using 50pin and 26pin bus.
4. Short 1 & 2 pins of JP9 in 8251 kit
5. Keep the DIP switch in 1 & 7 on (8086kit), open TALK, and go to options select target evice as 8086 and Connect.
6. Change dip switch into 1 & 5on, once reset 8086 kit.
7. Go to file ?Download hex file
8. Change the DIP switch into 1 & 7 on, once reset.
9. Remove RS232 cable from 8086 kit and connect it to 8251.
10. G-4000 (on 8086 kit keyboard) .enter
11. Give some input from system keyboard (Example press A, B, C, D enter),once reset 8086 kit That data will be received at 8086 kit at location FF00 (press E, enter address FF00 and press Comma to get the ASCII values of A, B, C,D).

PROGRAM

⇒System to Kit
ALS 8051/31 Monitor V1.0
8051>A
~~Enter PROGRAM EXECUTION...~~
Enter START Address = 8000
8000 MOV A,#03
8000 7403 MOV A,#03

8002 MOV B,#05
8002 ~~750~~ 750B65 MOV B,#05
8005 ADD A,B
8005 250B ADD A,B
8007 LCALL 03
8007 12D003 LCALL 003
800A
=> Kit to system

8051>
ALS 8051/31 Monitor V1.0
8051>G
PROGRAM EXECUTION...
Enter START Address = 8000

ACC	PSW	DPH	DPL	PCH	PCL	SP	B
07	01	00	00	04	60	00	00
R0	R1	R2	R3	R4	R5	R6	R7
00	00	00	00	00	00	00	00

RESULT

I have written an Assembly language Program to estoblish Communication between two Processing using 8251.