```
; program 1  (UP to Q)
data segment
        pa equ 0c800h
        pctrl equ 0c803h
data ends
code segment
assume cs:code , ds:data
        start:  mov ax,data
                mov ds,ax

                mov al,80h
                mov dx,pctrl
                out dx,al

                mov al,01
              top:mov dx,pa
                out dx,al

                mov bl,al

                mov ah,01h
                int 21h

                ;cmp al,81D
                cmp al,'Q'
                je stop

                call delay

                mov al,bl
                rol al,01
                jmp top

                delay proc near
                mov si,1234h
              t2: mov di,0ffffh
              t1: dec di
                jnz t1
                dec si
                jnz t2
                ret
                delay endp
             stop:mov ah,4ch
                int 21h
code ends
end start
```

; Program to count the number of 1's in a given number

```
data segment
 pa equ 0c800h
 pb equ 0c801h
 pctrl equ 0c803h
data ends
code segment
   assume cs:code, ds:data
       start:mov ax, data
             mov ds, ax

             mov al,82h
             mov dx,pctrl
             out dx,al

             mov dx,pb
             in al,dx

             mov cl,08h
             mov bl,00
          top:shr al,01
             jnc next_rot
             inc bl
    next_rot:dec cl
             jnz top
             mov al,bl
             mov dx,pa
             out dx,al
             mov ah,4ch
             int 21h
     code ends
     end start
```

```
; program BCD Counter

data segment
  pa equ 0c800h
  pctrl equ 0c803h
data ends

code segment
 assume cs:code,ds:data
    start:mov ax,data
          mov ds,ax

          mov al,80h
          mov dx,pctrl
          out dx,al

          mov al,00h
      top: mov dx,pa
          out dx,al
          call delay
          add al,01h
          daa
          cmp al,20h
          jle top
   bottom: sub al,1
          das
          mov dx,pa
          out dx,al
          call delay
          cmp al,00
          jz exit
          jmp bottom
     exit:mov ah,4ch
          int 21h


          delay proc
          mov bx,1234h
        t:mov cx,0ffffh
       loop1: loop loop1
         dec bx
         jnz t
         ret
         delay endp
          code ends
          end start
```

; left to right rolling fassion

```
data segment
pctrl equ 0c803h
pc equ 0c802h
pb  equ 0c801h
code1 db 0ffh,0ffh,0ffh,0ffh,99h,0b0h,0a4h,0f9h,80h,0f8h,82h,92h

data ends
code segment
     assume cs:code,ds:data
          start:mov ax,data
               mov ds,ax
               mov dx,pctrl
               mov al,80h
               out dx,al

               mov cl,12
               mov si,offset code1
          again:      call display
               call delay
               inc si
               dec cl
               jnz again
               mov ah,4ch
               int 21h

           display proc near
                mov bl,08h
               mov al,[si]
          top:   rol al,01
               mov ch,al
               mov dx,pb
               out dx,al
               mov al,00h
               mov dx,pc
               out dx,al
               mov al,0ffh
               mov dx,pc
               out dx,al
               mov al,ch
               dec bl
               jnz top
               ret
               display endp

              delay proc near
```

```
        push bx
        mov di,0ffffh
 t:     mov bx,0ffffh
 t1:    dec bx
        jnz t1
        dec di
        jnz t
        pop bx
        ret
delay endp
code ends
end start
```

```
; Seven segment
; Flickering effect  (1234 5678)
data segment
portc equ 0c802h
portb equ 0c801h
cw equ 0c803h
cod1 db 99h,0b0h,0a4h,0f9h
cod2 db 80h,0f8h,82h,92h
count db 5
data ends
  code segment
assume cs:code,ds:data
start:mov ax,data
      mov ds,ax

      mov al,80h
      mov dx,cw
      out dx,al

  again:lea si,cod1
        call display
        call delay
        lea si,cod2
        call display
        call delay
        dec count
        jnz again
        mov ah,4ch
        int 21h

          display proc
        mov di,0004
    nextchar:mov al,[si]
             mov bh,08
      nextbit:rol al,01
              mov cl,al
              mov dx,portb
              out dx,al
              mov al,00
              mov dx,portc
              out dx,al
              mov al,0ffh
              out dx,al
              mov al,cl
              dec bh
              jnz nextbit
```

```
   inc si
   dec di
   jnz nextchar
   ret
   display endp

    delay proc
    mov si,0ffffh
 t2:mov di,0bbbbh
t1:dec di
   jnz t1
   dec si
   jnz t2
   ret
   delay endp
    code ends
      end start
```

```
data segment
    pb equ 0cd01h
    pc equ 0cd02h
    pctrl equ 0cd03h
    tab db 1,0ah,0bh,0ch,0dh,0eh,0fh,2
    count db 8
    seg_tab db 0c0h,0ffh,0a4h,0ffh,99h,0ffh,82h,0ffh,80h,0ffh,88h,0ffh,0c6h,0ffh,86h,0ffh
data ends
code segment
assume cs:code,ds:data
start:  mov ax,data
        mov ds,ax
        mov al,80h
        mov dx,pctrl
        out dx,al
        mov cl,4
    clr:    mov al,0ffh
        call disp
        loop clr
        lea si,tab
        mov cl,count
     top:mov al,[si]
        shr al,1
        jc neven
        mov bx,offset seg_tab
        mov al,[si]
        xlat
        call disp
     neven:inc si
        dec cl
        jnz top
        mov ah,4ch
        int 21h

        disp proc near
        mov bl,8h
    again:rol al,01
        mov bh,al
        mov dx,pb
        out dx,al
        mov al,00h
        mov dx,pc
        out dx,al
        mov al,0ffh
        out dx,al
        mov al,bh
```

```
        dec bl
        jnz again
        ret
        disp endp
code ends
end start
```

;Seven segment
; pressing even digit gives only output
; otherwise (odd) no output

```
data segment
portc equ 0c802h
portb equ 0c801h
cw equ 0c803h
seg_table db 0c0h,0ffh,0a4h,0ffh,99h,0ffh,82h,0ffh,80h,0ffh
count db 5
data ends
  code segment
assume cs:code,ds:data
start:mov ax,data
      mov ds,ax
      mov ah,01h
      int 21h
      sub al,30h
      mov cl,al
      shr al,1
      jc noteven
      mov bx,offset seg_table
      mov al,cl
      xlat
      mov bl,al
      mov al,80h
      mov dx,cw
      out dx,al
      call display

   noteven:     mov ah,4ch
        int 21h
          display proc

          mov al,bl
         mov bh,08
       nextbit:rol al,01
              mov cl,al
              mov dx,portb
              out dx,al
              mov al,00
              mov dx,portc
              out dx,al
              mov al,0ffh
              out dx,al
              mov al,cl
```

```
dec bh
jnz nextbit

ret
display endp
 code ends
   end start
```

```
;stepper motor
 data segment

      n dw 0005
     pc equ 0c802h
     pctrl equ 0c803h
 data ends

 code segment
         assume cs:code,ds:data
          start: mov ax,data
                 mov ds,ax
                 mov al,80h  ;Move control word
                             ;to al
                 mov dx,pctrl
                 out dx,al   ;Contents of al is
                             ;moved to o/p port C
                 mov cx,n
                 mov al,0eeh

                 mov dx,pc
           t1:   out dx,al
                 call delay

                 rol al,1


                 dec cx
                 jnz t1
             mov cx,n
             mov al,77h
             mov dx,pc
         t2:  out dx,al
             call delay
             ror al,1
             dec cx
             jnz t2
                mov ah,4ch
                 int 21h

            delay proc near
            mov si, 0ffffh
       t4:    mov di,0ffffh
           t5:    dec di
                  jnz t5
                  dec si
```

```
                jnz t4
                ret
        delay endp
code ends
end start
```

; key scan

```
data segment
col db 00h
row db 00h
pa equ 0c800h
pc equ 0c802h
pctr1 equ 0c803h
key db 00h
newline db 0ah,0dh,'$'
data ends

code segment
 assume cs:code,ds:data
 start :mov ax,data
        mov ds,ax

        mov dx,pctr1
        mov al,90h
        out dx,al
        call keyscan
        mov row,bh
        call display
        mov dx,offset newline
        mov ah,09h
        int 21h
        mov ch,row
        inc ch

        call display
        mov dx,offset newline
        mov ah,09h
        int 21h
        mov ch,col
        inc ch
         call display
         mov dx,offset newline
         mov ah,09h
         int 21h
         mov ah,01h
         int 21h

         mov ah,4ch
         int 21h

         keyscan proc near
          repeat:mov bh,02h
```

```
        mov ch,10h
        mov bl,04h
        mov ah,00h

nextrow: mov al,bl
        mov dx,pc
        out dx,al

        ror bl,01
        mov dx,pa
        in al,dx
        cmp al,00h

        jnz findkey
        sub ch,08h

        dec bh
        cmp bh,0ffh
        jnz nextrow
        jmp repeat

 findkey: rcr al,01h
         jc keyfound
         inc col
         inc ch
         jmp findkey
 keyfound: ret
 keyscan endp

 display proc near
        mov dl,ch
        cmp dl,0ah
        jl asciiadd
        cmp dl,0fh
        jle t1
        mov bh,ch
        and ch,0f0h
        mov cl,04h
        ror ch,cl
        add ch,30h
        mov dl,ch
        mov ah,02h
        int 21h
        and bh,0fh
        add bh,30h
        mov dl,bh
```

```
  mov ah,02h
  int 21h
  jmp last

t1: add dl,07h
asciiadd:add dl,30h
  mov ah,02h
  int 21h

  last:ret

  display endp

  code ends
  end start
```

;Elevator

```
data segment
pctrl equ 0c803h
pa equ 0c800h
pb equ 0c801h
flor db 00,03,06,09,0e0h,0d3h,0b6h,79h
data ends
code segment
assume cs:code,ds:data
start:    mov ax,data
          mov ds,ax

          mov dx,pctrl
          mov al,82h
          out dx,al

          mov bl,00
          mov dx,pa
          mov al,bl
          or al,0f0h

          out dx,al    ;elevator in the ground floor
    top:  mov dx,pb
          in al,dx     ;check for the request
          or al,0f0h
          cmp al,0ffh
          jz top
 decide:  ror al,01    ;check from ehich floor the request has come
          jnc up
          inc si
          jmp decide
    up:   cmp bl,[si]  ;keep moving the ele until it reaches
          jz reset
          inc bl       ;the requested floor
          mov al,bl
          or al,0f0h
          mov dx,pa
          out dx,al
          call delay
          jmp up
reset:    add si,04    ;service the request
          mov al,[si]
          mov dx,pa
          out dx,al
          call delay
down:     dec bl       ;move ele down until it reaches ground floor
```

```
        cmp bl,0ffh
        jz stop
        mov al,bl
        or al,0f0h
        mov dx,pa
        out dx,al
        call delay
        jmp down
stop:   mov ah,4ch
        int 21h
     delay proc near
       mov cx,0ffffh
 t1:  mov di,0ffffh
 t:     dec di
        jnz t
        loop t1
         ret
         delay endp
         code ends
         end start
```

; key scan and divide

```
data segment
pa equ 0cd00h
pc equ 0cd02h
pctr1 equ 0cd03h
op1 db ?
op2 db ?
newline db 0ah,0dh,'$'
data ends

code segment
 assume cs:code,ds:data
 start :mov ax,data
        mov ds,ax

        mov dx,pctr1
        mov al,90h
        out dx,al

        call keyscan
        mov op1,ch
        call display

        mov ah,01h ; DOS interrupt to wait for the next
        int 21h    ; character from the keyboard

        call keyscan
        mov op2,ch
        call display

        mov al,op1
        mov ah,00
        div ch
        mov ch,al
        mov cl,ah   ; a copy of ah to cl
        call display

        mov ch,cl
        call display

        mov ah,4ch
        int 21h

     keyscan proc near
          repeat:mov bh,02h
```

```
        mov ch,10h
        mov bl,04h

  nextrow: mov al,bl
        mov dx,pc
        out dx,al

        ror bl,01
        mov dx,pa
        in al,dx
        cmp al,00h

        jnz findkey
        sub ch,08h

        dec bh
        cmp bh,0ffh
        jnz nextrow
        jmp repeat

  findkey: rcr al,01h
        jc keyfound
        inc ch
        jmp findkey
  keyfound: ret
keyscan endp

    display proc near
            mov dl,ch
            add dl,30h
            mov ah,02h
            int 21h

            mov dx,offset newline
            mov ah,09h
            int 21h

            ret
    display endp

          code ends
          end start
```

;Elevator

```
data segment
val1 db 03
val2 db 02

pctrl equ 0cd03h
pa equ 0cd00h
pb equ 0cd01h
flor db 00,03,06,09,0e0h,0d3h,0b6h,79h
data ends
code segment
assume cs:code,ds:data
start:    mov ax,data
          mov ds,ax
          mov dx,pctrl
          mov al,82h

          out dx,al
          lea si,flor

          mov bl,00
          mov dx,pa
          mov al,bl
          or al,0f0h
          out dx,al    ;elevator in the ground floor

          mov al,val1
          mul val2
          cmp al,00
          jz move
          inc si
          cmp al,03
          jz move
          inc si
          cmp al,06
          jz move
          inc si
          cmp al,09
          jz move
          jmp stop
    move:

         inc bl       ;the requested floor
         mov al,bl
         or al,0f0h
         mov dx,pa
```

```
        out dx,al
   ;    call delay
        cmp bl,[si]
        jnz move

        add si,04    ;service the request
        mov al,[si]
        mov dx,pa
        out dx,al
    ;   call delay
down:   dec bl       ;move ele down until it reaches ground floor
        cmp bl,0ffh
        jz stop
        mov al,bl
        or al,0f0h
        mov dx,pa
        out dx,al
     ;  call delay
        jmp down
 stop:  mov ah,4ch
        int 21h
      delay proc near
        mov cx,0ffffh
  t1:  mov di,0ffffh
  t:    dec di
        jnz t
        loop t1
         ret
         delay endp
         code ends
         end start
```

;Elevator

```
data segment
val1 db 03
val2 db 02

pctrl equ 0cd03h
pa equ 0cd00h
pb equ 0cd01h
flor db 00,03,06,09,0e0h,0d3h,0b6h,79h
data ends
code segment
assume cs:code,ds:data
start:    mov ax,data
          mov ds,ax
          mov dx,pctrl
          mov al,82h

          out dx,al
          lea si,flor

          mov bl,00
          mov dx,pa
          mov al,bl
          or al,0f0h
          out dx,al    ;elevator in the ground floor

          mov al,val1
          mul val2
          cmp al,00
          jz move
          inc si
          cmp al,03
          jz move
          inc si
          cmp al,06
          jz move
          inc si
          cmp al,09
          jz move
          jmp stop
    move:

         inc bl       ;the requested floor
         mov al,bl
         or al,0f0h
         mov dx,pa
```

```
        out dx,al
   ;    call delay
        cmp bl,[si]
        jnz move

        add si,04    ;service the request
        mov al,[si]
        mov dx,pa
        out dx,al
    ;   call delay
down:   dec bl       ;move ele down until it reaches ground floor
        cmp bl,0ffh
        jz stop
        mov al,bl
        or al,0f0h
        mov dx,pa
        out dx,al
     ;  call delay
        jmp down
 stop:  mov ah,4ch
        int 21h
      delay proc near
        mov cx,0ffffh
  t1:  mov di,0ffffh
  t:    dec di
        jnz t
        loop t1
         ret
         delay endp
         code ends
         end start
```

; key scan

```
data segment
pa equ 0cd00h
pc equ 0cd02h
pctr1 equ 0cd03h
msg db "PanduBabu$"
newline db 0ah,0dh,'$'
data ends

code segment
 assume cs:code,ds:data
 start :mov ax,data
        mov ds,ax

        mov dx,pctr1
        mov al,90h
        out dx,al

        call keyscan
        mov ah,02h
        int 21h
        lea si,msg
  again:call display
        mov dx,offset newline
        mov ah,09h
        int 21h
        dec ch
        jnz again
        mov ah,4ch
        int 21h

        keyscan proc near
         repeat:mov bh,02h
                mov ch,10h
                mov bl,04h

          nextrow: mov al,bl
                   mov dx,pc
                   out dx,al

                   ror bl,01
                   mov dx,pa
                   in al,dx
                   cmp al,00h

                   jnz findkey
```

```
        sub ch,08h

        dec bh
        cmp bh,0ffh
        jnz nextrow
        jmp repeat

findkey: rcr al,01h
         jc keyfound
         inc ch
         jmp findkey
keyfound: ret
keyscan endp

display proc near
          mov dx,offset msg
          mov ah,09h
          int 21h
          ret
       display endp

        code ends
        end start
```