1)

```
;blinking leds

back: mov p2,#00h
acall delay
mov p2,#0ffh
acall delay
sjmp back
delay: mov r0,#0ffh
        mov r1,#0ffh
        stay:djnz r0,stay
        djnz r1,stay
        ret
end
```

2);blinking leds by rolling from right to left and then left to right

```
mov a,#01h
mov r2,#0fh
left: acall delay
djnz r2,left
mov p2,a
rl a
jb p2.7,right
sjmp left

mov r3,#0fh


right: acall delay
djnz r3,right
rr a
mov p2,a
jb p2.0,left
sjmp right


delay: mov r0,#0ffh
        mov r1,#0ffh
        stay:djnz r0,stay
        stay1:djnz r1,stay1
        ret
end
```

3) rolling leds from right to left(msb to lsb)

```
mov a,#80h
mov r2,#0fh
back: acall delay
djnz r2,back
mov p2,a
rr a
sjmp back

delay: mov r0,#0ffh
        mov r1,#0ffh
        stay:djnz r0,stay
        stay1:djnz r1,stay1
        ret
end
```

4)

```
;blinking leds in even and odd positions      p7 p6 p5 p4     p3 p2 p1 p0  and rolling
                                                1  1            1  1

;blink leds in even and odd positions

program:

 mov a,#0AAh
back: mov p2,a
mov r2,#0fh

left: acall delay
djnz r2,left

rr a
sjmp back

delay: mov r0,#0ffh
        mov r1,#0ffh
        stay:djnz r0,stay
        djnz r1,stay
        ret
end
```

5) automatic water level indicator with buzzer

program:

```
main:    mov p2,#00h
         mov p1,#00h
         mov p3,#00h
         mov a,#00h
         clr p3.2

start: mov a,p1
     mov p2,a
    cjne a,#0ffh,MOTOR_ON
    setb p3.2
    acall delay
    clr p3.2
   acall delay
   clr p3.1
    clr p3.0

MOTOR_ON:cjne a,#01h,start
          clr p3.1
         setb p3.0
         sjmp start

delay:
        mov r2,#0ffh
       L1:djnz r2,L1
       ret

end
```

6) dispaly 00-99 numbers using two 7 segment dispaly

```
first:mov r1,#00h
    mov p2,r1

start: mov a,#00h
                  mov p3,a
back: acall delay
         cjne a,#0ah,next
         inc r1
         mov p2,r1
         acall delay
         cjne r1,#0ah,start
         sjmp first
next: inc a
         mov p3,a
         sjmp back
```

```
delay:  mov r0,#0ffh
                mov r2,#0ffh

stay:djnz r0,stay
        djnz r2,stay

        ret
        end
```

7) real time clock having hours-minutes

```
start:mov r6,#00h
mov r1,#00h
first:mov a,#00h
    mov r7,#00h

back: mov p2,a
     acall delay
          inc a
        inc r7
         cjne r7,#0ah,back
          add a,#06h
   mov r7,#00h
          cjne a,#60h,back
   inc r1
          inc r6
goto:  mov p3,r1
          acall delay
          cjne r6,#0ah,first
          mov r5,a
          mov a,r1
          add a,#06h
          mov r6,#00h
          mov r1,a
          mov a,r5
          cjne r1,#0dh,goto
          sjmp start

delay:mov r0,#0ffh
    mov r2,#0ffh
stay: djnz r0,stay
          ;djnz b,stay
          djnz r2,stay
          ret
end
```

ANOTHER METHOD

```
MOV R7,#02
START: MOV B,#00H
```

```
BACK1:MOV R6,#13D
BACK2: MOV A,B
       MOV P3,A
       MOV A,#00H
       MOV R4,#06H
BACK3: MOV R3,#0AH
BACK4: MOV P2,A
ACALL DELAY
INC A
DJNZ R3,BACK4
ADD A,#06H
DJNZ R4,BACK3
INC B
MOV A,B
ADD A,#00H
DA A
MOV B,A
DJNZ R6,BACK2
MOV B,#01H
MOV R6,#11D
DJNZ R7,BACK2
SJMP START
DELAY: MOV R0,#0FFH
     MOV R1,#0FFH
       MOV R2,#01H
L1:    DJNZ R0,L1
```

```
    DJNZ R1,L1

        DJNZ R2,L1

        RET
END
```

7)Home automation

```
MOV TMOD,#10H
MOV TH1,#00H
MOV TL1,#00H
SETB TR1

M EQU P1.0
N EQU P1.1
T EQU P2.0
L EQU P1.2

MOV P1,#0FFH
MOV P2,#00H

START: JNB L,LAMP_OFF
JNB M,ENTER
JNB N,LEAVE
SJMP START

ENTER: JNB N,LAMP_ON
SJMP ENTER

LAMP_ON:setb t
SETB M
SETB N
ACALL DELAY
SJMP START
LEAVE: JNB M,LAMP_OFF
SJMP LEAVE
LAMP_OFF:
CLR T
SETB M
SETB N
ACALL DELAY
SJMP START
```

```
DELAY:JNB TF1,DELAY
CLR TF1
RET

END
```

8) using pushbuttons and lcd counting visitors in a mall.

```
MOV TMOD,#10H
MOV TH1,#00H
MOV TL1,#00H
mov r6,#00h

RS EQU P1.0
RW EQU P1.1
EN EQU P1.2
LCD EQU P2

MOV A,#38H
ACALL  CMD
MOV A,#06H
ACALL  CMD
MOV A,#0EH
ACALL  CMD
MOV A,#01H
ACALL CMD
MOV A,#80H
ACALL  CMD

MOV DPTR,#TEXT1
BACK: CLR A
MOVC  A,@A+DPTR
JZ STOP
ACALL SEND
INC DPTR
SJMP BACK

STOP:
M EQU P1.3
N EQU P1.4
MOV R7,#00H
SETB M
SETB N

CHECK:

JNB M,ENTER
JNB N,LEAVE
SJMP CHECK
```

```
ENTER: JNB N,INR
SJMP ENTER

INR: INC R6
MOV A,R6
ACALL HEX
MOV A,#0C0H
ACALL SEE
INC R7
MOV A,R7
ACALL HEX
MOV A,#0CAH
ACALL SEE
SETB M
SETB N
SJMP CHECK

LEAVE: JNB M,DER
SJMP LEAVE

DER: DEC R7
MOV A,R7
ACALL HEX
MOV A,#0CAH
ACALL SEE
SETB M
SETB N
SJMP CHECK

CMD: MOV LCD,A
CLR RS
CLR RW
SETB EN
ACALL DELAY
CLR EN
RET

SEND: MOV LCD,A
SETB RS
CLR RW
SETB EN
ACALL DELAY
CLR EN
RET

GET:
ANL A,#0FH
ACALL ASCII
ACALL SEND
RET

HEX:
```

```
MOV B,#0AH
DIV AB
XCH A,B
MOV R0,A
XCH A,B
MOV B,#0AH
DIV AB
XCH A,B
MOV R1,A
XCH A,B
MOV R2,A
RET

ASCII: MOV R5,A
CLR C
SUBB A,#0AH
MOV A,R5
JC SKIP
ADD A,#07H
SKIP: ADD A,#30H
RET

SEE:ACALL CMD
MOV A,R2
ACALL GET
MOV A,R1
ACALL GET
MOV A,R0
ACALL GET



DELAY:SETB TR1
WAIT:JNB TF1,WAIT
CLR TR1
CLR TF1
RET

ORG 1000H
        TEXT1: DB 'TOTAL:::::NET',00H
END
```