```
JMP st1
db 5 dup(0)

;IVT entry for NMI (INT 02h)
DW Nmi_24hrtimer
DW 0000

DB 500 dup(0)

;IVT entry for 80h
DW Switch_intR
DW 0000
DB 508 dup(0)
st1: CLI

;Initialise DS,ES,SS to start of RAM
 MOV AX,0100h ;;;;;;;;;;;;;;;;;;;;
 MOV DS,AX
 MOV ES,AX
 MOV SS,AX
 MOV SP,0FFFEh

;Initialise 8255
 STI
 ;8255-1
 MOV AL,88h ;;;;;CHANGE THIS
 OUT 06h,AL
 ;8255-2
 MOV AL,88h ;;;;;CHANGE THIS IF REQUIRED
 OUT 0Eh,AL

;Initialise Timers
 MOV AL,36h ;Control Word for 8253-1 C0
 OUT 16h,AL

 MOV AL,56h ;Control Word for 8253-1 C1
 OUT 16h,AL

 MOV AL,94h ;Control Word for 8253-1 C2
 OUT 16h,AL

 MOV AL,14h ;Control Word for 8253-2 C0
 OUT 1Eh,AL

 MOV AL,54h ;Control Word for 8253-2 C1
 OUT 1Eh,AL

 MOV AL,94h ;Control Word for 8253-2 C2
 OUT 1Eh,AL

 MOV AL,14h ;Control Word for 8253-3 C1
```

```
OUT 26h,AL

MOV AL,0A8h ;Load count lsb for 8253-1 C0
OUT 10h,AL

MOV AL,61h ;Load count msb for 8253-1 C0
OUT 10h,AL

MOV AL,64h ;Load count lsb for 8253-1 C1
OUT 12h,AL

MOV AL,3Dh ;Load count lsb for 8253-1 C2 (1 Minute Timer)
OUT 14h,AL

MOV AL,3Ch ;Load count lsb for 8253-2 C0 (1 Min counter for 24 hr generation)
OUT 18h,AL

MOV AL,3Ch ;load count for 8253-2 C1 (1 hr counter for 24 hr generation)
OUT 1Ah,AL

MOV AL,18h ;Load count lsb for 8253-2 C2
OUT 1Ch,AL

MOV AL,2h ;Load count lsb for 8253-3 C1
OUT 20h,AL

MOV AL,00h ;default low output from 8255-2 upper port C
OUT 0Ch,AL

;LCD Initialisation
CALL DELAY_20ms
MOV AL,04h ;PB2 bit is 1 to make E on LCD 1
OUT 0Ah,AL ;;;;;;;;;;;;;

CALL DELAY_20ms
MOV AL,00h ;Make E on LCD 0
OUT 0Ah,AL ;;;;;;;;;;;;

MOV AL,38h ;8-bit interface + Two Lines + 5x7 dots
OUT 08h,AL

MOV AL,04h ;PB2 bit is 1 to make E on LCD 1
OUT 0Ah,AL ;;;;;;;;;;;;;

CALL DELAY_20ms
MOV AL,00h ;Make E on LCD 0
OUT 0Ah,AL ;;;;;;;;;;;;

CALL DELAY_20ms
MOV AL,0Ch ;Display On
OUT 08h,AL
```

```
 MOV AL,04h ;PB2 bit is 1 to make E on LCD 1
 OUT 0Ah,AL ;;;;;;;;;;;;;

 CALL DELAY_20ms
 MOV AL,00h ;Make E on LCD 0
 OUT 0Ah,AL ;;;;;;;;;;;;

 MOV AL,06h ;Increment
 OUT 08h,AL

 CALL DELAY_20ms
 MOV AL,04h ;PB2 bit is 1 to make E on LCD 1
 OUT 0Ah,AL ;;;;;;;;;;;;;

 CALL DELAY_20ms
 MOV AL,00h ;Make E on LCD 0
 OUT 0Ah,AL ;;;;;;;;;;;;

 MOV AL,4Ch ;Set CG RAM Adress
 OUT 08h,AL

 CALL DELAY_20ms
;End of LCD Initialisation

 mov si,0000h
 mov al,0bdh ;hard coding pass-word
    ;9999999999999999
 mov [si],al
 mov al,0bdh
 mov [si+1],al
 mov al,0bdh
 mov [si+2],al
 mov al,0bdh
 mov [si+3],al
 mov al,0bdh
 mov [si+4],al
 mov al,0bdh
 mov [si+5],al
 mov al,0bdh
 mov [si+6],al
 mov al,0bdh
 mov [si+7],al
 mov al,0bdh
 mov [si+8],al
 mov al,0bdh
 mov [si+9],al
 mov al,0bdh
 mov [si+0ah],al
 mov al,0bdh
```

```
 mov [si+0bh],al
 mov al,0bdh
 mov [si+0ch],al
 mov al,0bdh
 mov [si+0dh],al
 mov al,0bdh
 mov [si+0eh],al
 mov al,0bdh
 mov [si+0fh],al

 add si,000fh
 inc si
 mov al,0bdh ;hard coding alarm pass-word
    ;99999999999999
 mov [si],al
 mov al,0bdh
 mov [si+1],al
 mov al,0bdh
 mov [si+2],al
 mov al,0bdh
 mov [si+3],al
 mov al,0bdh
 mov [si+4],al
 mov al,0bdh
 mov [si+5],al
 mov al,0bdh
 mov [si+6],al
 mov al,0bdh
 mov [si+7],al
 mov al,0bdh
 mov [si+8],al
 mov al,0bdh
 mov [si+9],al
 mov al,0bdh
 mov [si+0ah],al
 mov al,0bdh
 mov [si+0bh],al
 mov al,0bdh
 mov [si+0ch],al
 mov al,0bdh
 mov [si+0dh],al
 add si,000dh

 inc si

 MOV AL,0FFh ;Initialise Port A of 8255-1
 OUT 00h,AL;;;;;;;;;;;;;

start: CALL clear_LCD
  CALL Welcome_msg
```

```
  MOV BP,00h ;Initialise
  CALL keypad_input
  CMP AL,7Eh
  JZ master_mode
  JMP start

x6: CALL clear_LCD
 CALL welcome_msg
 CALL keypad_input
 CMP AL,7Dh ;O key
 JZ User_mode
 JMP x6

master_mode: CALL entm
     MOV BP,0ABCDh
     CMP AX,0ABCDh
     JNZ x6

x8: CALL keypad_input
 CMP AL,7Bh ;A key
 JZ Alarm_mode
 JNZ x8

Alarm_mode: CALL enta
    CMP DH,6h ;Alarm caused by wrong Master pwd
    JZ start

    CMP DH,1h ;Alarm caused by wrong user pwd
    JZ x6
    JMP x70

User_mode: CALL entu
    CMP AX,0ABCDh
    JZ x8
    JNZ x6

x70: stop: JMP stop




;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;-------PROCEDURES--------;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 DELAY_20ms proc
  MOV CH,5h
  x4: NOP
   NOP
   DEC CH
   JNZ x4
```

```
 RET
DELAY_20ms ENDP

DELAY_max proc
 MOV CX,0FFFFh
 x16: NOP
   NOP
   DEC CX
   JNZ x16
 RET
DELAY_max ENDP

clear_LCD proc
 MOV AL,00h
 OUT 0Ah,AL ;;;;;;;;;;;

 CALL DELAY_20ms
 MOV AL,01h ;Clears Display
 OUT 08h,AL ;;;;;;;;;;;;;;

 CALL DELAY_20ms
 MOV AL,04h
 OUT 0Ah,AL ;;;;;;;;;;;

 CALL DELAY_20ms
 MOV AL,00h
 OUT 0Ah,AL ;;;;;;;;;;;

 RET
clear_LCD endp

welcome_msg proc
 MOV AL,0A0h
 OUT 08h,AL ;;;;;;;;;;;;;;;
 CALL DELAY_20ms
 MOV AL,05h
 OUT 0Ah,AL ;;;;;;;;;;;;
 CALL DELAY_20ms
 MOV AL,01h
 OUT 0Ah,AL ;Prints Space ;;;;;;;;;;;;;;

 MOV AL,57h
 OUT 08h,AL ;;;;;;;;;;;;;;
 CALL DELAY_20ms
 MOV AL,05h
 OUT 0Ah,AL ;;;;;;;;;;;;
 CALL DELAY_20ms
 MOV AL,01h
 OUT 0Ah,AL ;Prints W ;;;;;;;;;;;;;;;

 MOV AL,45h
```

```
OUT 08h,AL ;;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL ;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Prints E ;;;;;;;;;;;;;;;

MOV AL,4Ch
OUT 08h,AL ;;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL ;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Prints L ;;;;;;;;;;;;;;;

MOV AL,43h
OUT 08h,AL ;;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL ;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Prints C ;;;;;;;;;;;;;;;

MOV AL,4Fh
OUT 08h,AL ;;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL ;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Prints O ;;;;;;;;;;;;;;;

MOV AL,4Dh
OUT 08h,AL ;;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL ;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Prints M ;;;;;;;;;;;;;;;

MOV AL,45h
OUT 08h,AL ;;;;;;;;;;;;;;
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL ;;;;;;;;;;;;;
CALL DELAY_20ms
```

```
 MOV AL,01h
 OUT 0Ah,AL ;Prints E ;;;;;;;;;;;;;;;

 RET
welcome_msg ENDP

updateday_msg proc
 mov al,55h
 out 08h,al
 call DELAY_20ms
 mov al,05h
 out 0Ah,al
 call DELAY_20ms
 mov al,01h
 out 0Ah,al ;prints U

 mov al,50h
 out 08h,al
 call DELAY_20ms
 mov al,05h
 out 0Ah,al
 call DELAY_20ms
 mov al,01h
 out 0Ah,al ;prints P

 mov al,44h
 out 08h,al
 call DELAY_20ms
 mov al,05h
 out 0Ah,al
 call DELAY_20ms
 mov al,01h
 out 0Ah,al ;prints D

 mov al,41h
 out 08h,al
 call DELAY_20ms
 mov al,05h
 out 0Ah,al
 call DELAY_20ms
 mov al,01h
 out 0Ah,al ;prints A

 mov al,54h
 out 08h,al
 call DELAY_20ms
 mov al,05h
 out 0Ah,al
 call DELAY_20ms
 mov al,01h
```

```
 out 0Ah,al ;prints T

 mov al,45h
 out 08h,al
 call DELAY_20ms
 mov al,05h
 out 0Ah,al
 call DELAY_20ms
 mov al,01h
 out 0Ah,al ;prints E

 RET
updateday_msg ENDP

Print_* proc
 MOV AL,2Ah
 OUT 08h,AL
 CALL DELAY_20ms
 MOV AL,05h
 OUT 0Ah,AL
 CALL DELAY_20ms
 MOV AL,01h
 OUT 0Ah,AL ;Prints *;;;;;;;;;;;;;
 RET
Print_* ENDP

error_msg proc
 MOV AL,0A0h
 OUT 08h,AL
 CALL DELAY_20ms

 MOV AL,05h
 OUT 0Ah,AL
 CALL DELAY_20ms

 MOV AL,01h
 OUT 02h,AL ;Print Space

 MOV AL,49h
 OUT 08h,AL
 CALL DELAY_20ms
 MOV AL,05h
 OUT 0Ah,AL
 CALL DELAY_20ms
 MOV AL,01h
 OUT 0Ah,AL ;Print I

 MOV AL,4Eh
 OUT 08h,AL
 CALL DELAY_20ms
```

```
MOV AL,05h
OUT 0Ah,AL
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Print N

MOV AL,56h
OUT 08h,AL
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Print V

MOV AL,41h
OUT 08h,AL
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Print A

MOV AL,4Ch
OUT 08h,AL
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Print L

MOV AL,49h
OUT 08h,AL
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Print I

MOV AL,44h
OUT 08h,AL
CALL DELAY_20ms
MOV AL,05h
OUT 0Ah,AL
CALL DELAY_20ms
MOV AL,01h
OUT 0Ah,AL ;Print D
```

```
 RET
error_msg ENDP


clear_1digit_LCD proc ;Shift left -> print Space -> Shift Left
 MOV AL,00h
 OUT 0Ah,AL
 CALL DELAY_20ms

 MOV AL,10h
 OUT 08h,AL ;Shift Left by 1
 CALL DELAY_20ms
 MOV AL,04h
 OUT 0Ah,AL ;Enable High
 CALL DELAY_20ms
 MOV AL,00h ;Enable Low
 OUT 0Ah,AL

 MOV AL,0A0h
 OUT 08h,AL
 CALL DELAY_20ms
 MOV AL,05h
 OUT 0Ah,AL
 CALL DELAY_20ms
 MOV AL,01h
 OUT 0Ah,AL ;Prints Space

 CALL DELAY_20ms
 MOV AL,10h
 OUT 08h,AL
 CALL DELAY_20ms
 MOV AL,04h
 OUT 0Ah,AL
 CALL DELAY_20ms
 MOV AL,00h
 OUT 0Ah,AL

 RET
clear_1digit_LCD ENDP

keypad_input proc ;AL will contain the key value

x0: MOV AL,00h ;Check for Key release
 OUT 0Ch,AL ;;;;;;;;;;;;;;;;
x1: IN AL,0Ch ;;;;;;;;;;;;;;;;
 AND AL,0F0h
 CMP AL,0F0h
 JNZ x1
 CALL DELAY_20ms ;Debounce
```

```
 MOV AL,00h ;Check for Key press
 OUT 0Ch,AL ;;;;;;;;;;;;;;;
x2: IN AL,0Ch ;;;;;;;;;;;;;;
 AND AL,0F0h
 CMP AL,0F0h
 JZ x2
 CALL DELAY_20ms

 MOV AL,00h ;Check for Key press
 OUT 0Ch,AL ;;;;;;;;;;;;;
 IN AL,0Ch ;;;;;;;;;;;;;
 AND AL,0F0h
 CMP AL,0F0h
 JZ x2

 MOV AL,0Eh ;Check for Key Press Column 1
 MOV BL,AL
 OUT 0Ch,AL ;;;;;;;;;;;;;;;;;;;;;;;;
 IN AL,0Ch ;;;;;;;;;;;;;;;;;
 AND AL,0F0h
 CMP AL,0F0h
 JNZ x3
 MOV AL,0Dh ;Check for Key Press Column 2
 MOV BL,AL
 OUT 0Ch,AL ;;;;;;;;;;;;;;;;;;;;;;;;;;
 IN AL,0Ch ;;;;;;;;;;;;;;;;;;;;;;
 AND AL,0F0h
 CMP AL,0F0h
 JNZ x3
 MOV AL,0Bh ;Check for Key Press Column 3
 MOV BL,AL
 OUT 0Ch,AL ;;;;;;;;;;;;;;;;;;;;;;;;
 IN AL,0Ch ;;;;;;;;;;;;;;;;;;;;
 AND AL,0F0h
 CMP AL,0F0h
 JNZ x3
 MOV AL,07h ;Check for Key Press Column 4
 MOV BL,AL
 OUT 0Ch,AL ;;;;;;;;;;;;;;;;;;;;;;;;
 IN AL,0Ch ;;;;;;;;;;;;;;;;;;;;
 AND AL,0F0h
 CMP AL,0F0h
 JZ x2

x3: OR AL,BL

 RET
keypad_input ENDP
```

```
close_door proc
 MOV AL,83h
 OUT 02h,AL

 CALL DELAY_max

; x100: IN AL,04h ;Input from Timer
;   CMP AL,03h
;   JNZ x100

 MOV AL,00h
 OUT 02h,AL

 RET
close_door ENDP

open_door proc
 CALL clear_LCD
 MOV AL,8Ah
 OUT 02h,AL

 CALL DELAY_20ms

; MOV AL,0Ah
; OUT 02h,AL

x31: IN AL,04h ; Input from Timer
  CMP AL,0F0h
  JNZ x31

 CALL DELAY_20ms
 CALL close_door
 RET
open_door ENDP

entm proc
 CALL clear_LCD
 MOV AL,0FEh
 OUT 00h,AL  ;Turn On Enter Pwd LED
 MOV CX,16

enter_16_bit: CALL keypad_input

    CMP AL,0BBh
    JZ press_c

    CMP AL,0B7h
    JZ press_ac

    CMP AL,77h
    JZ press_enter
```

```
    CMP AL,7Eh
    JZ invalid_master ;Invalid key pressed like M,O,A

    CMP AL,7Dh
    JZ invalid_master

    CMP AL,7Bh
    JZ invalid_master

    MOV [SI],AL
    CALL Print_*

    INC SI
    DEC CX
    JNZ enter_16_bit

disp_enter_master: CALL keypad_input
     CMP AL,0BBh
     JZ press_c

     CMP AL,0B7h
     JZ press_ac

     CMP AL,77h
     JZ press_enter;

     JMP disp_enter_master ;;;;;;;;;;;;;;;;;

invalid_master: NOP
    JMP enter_16_bit

press_c: CALL clear_1digit_LCD
   DEC SI
   INC CX
   JMP enter_16_bit

press_ac: CALL clear_LCD
   MOV CX,16
   MOV SI,1Eh
   JMP enter_16_bit

press_enter: CALL clear_LCD
    MOV AL,0FFh ;Turn Off all LEDs
    OUT 00h,AL
    CMP CX,0
    JZ cmp_pass
    JMP raise_alarm

day_pass: MOV SI,002Eh
```

```
    MOV AL,0FDh
    OUT 00h,AL ;Turn On Retry/Update LED

    CALL DELAY_max
    CALL DELAY_max
    CALL DELAY_max

    CALL clear_LCD
    MOV CX,12

enter_12_bit_m: CALL keypad_input
     CMP AL,0BBh
     JZ press_cday

     CMP AL,0B7h
     JZ press_acday

     CMP AL,7Eh
     JZ invalid_day

     CMP AL,7Dh
     JZ invalid_day

     CMP AL,7Bh
     JZ invalid_day

     CMP AL,77h
     JZ press_enter_day

     MOV [SI],AL
     CALL Print_*
     INC SI
     DEC CX
     JNZ enter_12_bit_m

disp_enter: CALL keypad_input
    CMP AL,0BBh
    JZ press_cday

    CMP AL,0B7h
    JZ press_acday

    CMP AL,77h
    JZ press_enter_day

    JMP disp_enter ;;;;;;;;;;;;;;;;

invalid_day: NOP
     JMP enter_12_bit_m

press_cday: CALL clear_1digit_LCD
```

```
   DEC SI
   INC CX
   JMP enter_12_bit_m

press_acday: CALL clear_LCD
    MOV CX,12
    MOV SI,002Eh
    JMP enter_12_bit_m

press_enter_day: CALL clear_LCD
     MOV AL,0FFh
     OUT 00h,AL ; Shut down all LEDs

     CMP CX,0
     JNZ err_msg

     MOV AL,0Fbh
     OUT 00h,AL ; Turn On Pwd Updated LED

     CALL DELAY_max
     CALL DELAY_max

     MOV AL,0FFh
     OUT 08h,AL
     JZ end_69h

err_msg: CALL error_msg
   JMP day_pass

cmp_pass: CLD
   MOV SI,0000h
   MOV DI,001Eh
   MOV CX,17

  x5: MOV AL,[SI]
   MOV BL,[DI]
   DEC CX
   JZ day_pass ; Master Pwd is Correct...Proceed to set new pwd
   CMP AL,BL
   JNZ raise_alarm
   INC SI
   INC DI
   JMP x5

raise_alarm: MOV DH,5h
    MOV AL,0Fh ;Turn On Alarm and turn off the 3 LEDs
    OUT 00h,AL
    MOV AX,0ABCDh

end_69h: RET
```

```
entm ENDP

enta proc
 MOV AL,0Eh
 OUT 00h,AL ;Glow Enter Pwd LED

 MOV CX,14
 MOV SI,3Ah

enter_14_bit: CALL keypad_input
    CMP AL,0BBh
    JZ press_c_alarm

    CMP AL,0B7h
    JZ press_ac_alarm

    CMP AL,7Eh
    JZ invalid_alarm

    CMP AL,7Dh
    JZ invalid_alarm

    CMP AL,7Bh
    JZ invalid_alarm

    CMP AL,77h
    JZ press_enter_alarm

    MOV [SI],AL
    CALL Print_*
    INC SI
    DEC CX
    JNZ enter_14_bit

disp_enter_alarm: call keypad_input
     CMP AL,0BBh
     JZ press_c_alarm

     CMP AL,0B7h
     JZ press_ac_alarm

     CMP AL,77h
     JZ press_enter_alarm

invalid_alarm: NOP
    JMP enter_14_bit

press_c_alarm: CALL clear_1digit_LCD
    DEC SI
    INC CX
```

```
    JMP enter_14_bit

press_ac_alarm: CALL clear_LCD
    MOV CX,14
    MOV SI,3Ah
    JMP enter_14_bit

press_enter_alarm: CALL clear_LCD
     MOV AL,0Fh
     OUT 00h,AL ;Turn off LEDs

     CMP CX,0
     JZ cmp_pass_alarm

     JNZ x56

cmp_pass_alarm: CLD
    MOV SI,10h
    MOV DI,3Ah
    MOV CX,14
    REPE CMPSB
    CMP CX,00h
    JNZ x56
    MOV AL,0FFh
    OUT 00h,AL
    ADD DH,1h

x56: RET

enta ENDP

entu proc
 CALL clear_LCD
 MOV DL,1 ; For checking two inputs

 MOV AL,0FEh
 OUT 00h,AL ;Turn on enter pwd LED

 MOV CX,12
 MOV SI,48h

enter_12_bit: CALL keypad_input
    CMP AL,0BBh
    JZ press_c_user

    CMP AL,0B7h
    JZ press_ac_user

    CMP AL,7Eh
    JZ invalid_user
```

```
    CMP AL,7Dh
    JZ invalid_user

    CMP AL,7Bh
    JZ invalid_user

    CMP AL,77h
    JZ press_enter_user

    MOV [SI],AL
    CALL Print_*
    INC SI
    DEC CX
    JNZ enter_12_bit

disp_enter_user: CALL keypad_input
     CMP AL,0BBh
     JZ press_c_user

     CMP AL,0B7h
     JZ press_ac_user

     CMP AL,77h
     JZ press_enter_user

invalid_user: NOP
    JMP enter_12_bit

press_c_user: CALL clear_1digit_LCD
    DEC SI
    INC CX
    JMP enter_12_bit

press_ac_user: CALL clear_LCD
    MOV CX,12
    MOV SI,48h
    JMP enter_12_bit

press_enter_user: MOV AL,0FFh
     OUT 00h,AL
     CMP CX,0h
     JZ cmp_pass_user
     JNZ wrong_pass

cmp_pass_user: CLD
    MOV SI,2Eh
    MOV DI,48h
    MOV CX,12
    REPE CMPSB
    CMP CX,00h
```

```
    JNZ wrong_pass
    JZ open_door_user

wrong_pass: CALL clear_LCD
   MOV SI,48h
   MOV CX,12
   CMP dl,0
   JZ raise_alarm_user
   MOV AL,0FDh
   OUT 00h,AL

   DEC DL
   JMP enter_12_bit

raise_alarm_user: MOV DH,0h
     MOV AL,0Fh
     OUT 00h,AL
     MOV AX,0ABCDh
     JMP end_70h

open_door_user: CALL open_door

end_70h: RET

entu ENDP

Nmi_24hrtimer: CALL clear_LCD
    CALL clear_1digit_LCD
    CALL updateday_msg

 startnmi: CALL keypad_input
    CMP AL,07Eh
    JZ master_mode
    JMP startnmi

IRET

Switch_intR: CALL open_door
    STI
    CMP BP,0ABCDh
    JZ x6
    JNZ start
```