

Δανάη Σπέντζου 03120237
Νεκτάριος Μπούμπαλος 03120441

1η Άσκηση

IN 10H

```
MVI A, 10H      ; Set up Display
STA 0B00H
STA 0B01H
STA 0B02H
STA 0B03H
STA 0B04H
STA 0B05H
```

```
MVI A, 0DH      ; Initialization of Interrupt mask
SIM
EI
```

```
WAIT:
JMP WAIT
```

```
INTR_ROUTINE:
POP H           ; POP return address so that the stack doesn't fill up
EI             ; Enable interrupts inside interrupt routine
MVI A, 00H      ; Turn on LEDs
STA 3000H
```

```
MVI H, 06H      ; Counter for 6 iterations
MOV A, H
DCR A           ; Set up tens
STA 0B04H       ; Store tens in the 4th segment display
```

```
SECONDS:
MVI A, 09H      ; Set up 9 secs (units)
LIGHTS_ON:
STA 0B05H       ; Store units in the 5th segment display
CALL DISPLAY
DCR A
CPI 00H         ; Compare with zero
JNZ LIGHTS_ON  ; If Z=0 then 9 seconds passed
CALL ZERO      ; Display zero unit (1 sec)
DCR H          ; Decrease counter
JZ EXIT        ; If Z=0 end timer
MOV A, H
DCR A
STA 0B04H
JMP SECONDS     ; Repeat for 60 seconds
```

```
EXIT:
MVI A, FFH      ; Turn off LEDs
STA 3000H
JMP WAIT        ; Return to wait (main program)
```

```
DISPLAY:
LXI B, 0064H    ; 100 msec delay
LXI D, 0B00H    ; For STDM
PUSH PSW
PUSH H
PUSH D
PUSH B
CALL STDM
MVI A, 0AH      ; 10 * 100msec = 1 sec
```

```

1SEC:
CALL DCD
CALL DELB
DCR A
CPI 00H
JNZ 1SEC
POP B
POP D
POP H
POP PSW
RET

ZERO:                ; Display zero in the 5th segment display
MVI A, 00H
STA 0B05H
CALL DISPLAY
CALL DELB
RET

END

```

2η Άσκηση

```

IN 10H

MVI A, 10H           ; Set up Display
STA 0B00H
STA 0B01H
STA 0B03H           ; Update to the 4th segment display
STA 0B04H           ; Update to the 5th segment display
STA 0B05H

MVI A, 0DH           ; Initialization of Interrupt mask
SIM
EI

WAIT:
JMP WAIT

INTR_ROUTINE:
POP H                ; POP return address so that the stack doesn't fill up
EI                   ; Enable interrupts inside interrupt routine
MVI A, 00H           ; Turn on LEDs
STA 3000H

MVI H, 06H           ; Counter for 6 iterations
MOV A, H
DCR A                ; Set up tens
STA 0B04H           ; Store tens in the 5th segment display

SECONDS:
MVI A, 09H           ; Set up 9 secs (units)
LIGHTS_ON:
STA 0B05H           ; Store units in the 6th segment display
CALL DISPLAY
DCR A
CPI 00H              ; Compare with zero
JNZ LIGHTS_ON        ; If Z=0 then 9 seconds passed
CALL ZERO             ; Display zero unit (1 sec)
DCR H                 ; Decrease counter
JZ EXIT              ; If Z=0 end timer

```

```

MOV A, H
DCR A
STA 0B04H
JMP SECONDS      ; Repeat for 60 seconds

EXIT:
MVI A, FFH      ; Turn off LEDs
STA 3000H
JMP WAIT        ; Return to wait (main program)

DISPLAY:
LXI B, 0064H    ; 100 msec delay
LXI D, 0B00H    ; For STD
PUSH PSW
PUSH H
PUSH D
PUSH B
CALL STD
MVI A, 0AH      ; 10 * 100msec = 1 sec
1SEC:
CALL DCD
CALL DELB
DCR A
CPI 00H
JNZ 1SEC
POP B
POP D
POP H
POP PSW
RET

ZERO:           ; Display zero in the 6th segment display
MVI A, 00H
STA 0B05H
CALL DISPLAY
CALL DELB
RET

END

```

3η Άσκηση

α)

```

SWAP Nibble MACRO Q
    PUSH PSW
    MOV A, M
    RRC
    RRC
    RRC
    RRC
    MOV M, A
    MOV A, Q
    RLC
    RLC
    RLC
    RLC
    MOV Q, A
    POP PSW
ENDM

```

β)

```

FILL MACRO RP, X, K
    PUSH PSW
    PUSH
    MOV H, R
    MOV L, P
LOOP:
    MOV M, K
    INX H
    DCR X
    JNZ LOOP
    POP H
    POP PSW

    ENDM

```

```

γ)
RHLR MACRO
    PUSH PSW
    PUSH H
    MOV A, H
    RRC
    MOV A, L
    RRC

    POP H
    POP PSW

    ENDM

```

4η Άσκηση

Αρχικά ο μετρητής προγράμματος είναι (PC) = 0840H και ο δείκτης σωρού (SP) = 3000H. Κατά την εκτέλεση της εντολής CALL 0900H, προκαλείται hardware διακοπή RST 5.5, οπότε θα πάρει προτεραιότητα η εξυπηρέτησή της σε σχέση με το υπόλοιπο πρόγραμμα. Θα έχουμε, λοιπόν, τις ακόλουθες λειτουργίες:

1. Ολοκληρώνεται η εκτέλεση της εντολής CALL και αποθηκεύεται στην στοίβα διεύθυνση της εντολής που βρίσκεται μετά το CALL (η 0843H), καθώς η εντολή αυτή έχει μέγεθος 3 bytes. Ο PC παίρνει την τιμή 0900H. $(SP) = (SP) - 2H = 2FFE H$.
2. Στη συνέχεια, ενεργοποιείται η ρουτίνα εξυπηρέτησης διακοπής RST 5.5 και ο PC παίρνει ως τιμή την διεύθυνση 002CH. Ακόμη, αποθηκεύεται στη στοίβα η διεύθυνση 0900H. $(SP) = (SP) - 2H = 2FFCH$.
3. Αφού τελειώσει η ρουτίνα εξυπηρέτησης της RST 5.5, γίνεται POP από την στοίβα η προηγούμενη διεύθυνση του PC (δηλαδή η 0900H) και ο PC παίρνει την τιμή αυτή. $(SP) = (SP) + 2 = 2FFE H$.
4. Τέλος, ολοκληρώνεται η εκτέλεση της τελευταίας ρουτίνας στην διεύθυνση 0900H και γίνεται POP η διεύθυνση 0803H από την στοίβα και ο PC λαμβάνει την τιμή αυτή. $(SP) = (SP) + 2 = 3000 H$.
5. Συνεχίζεται η εκτέλεση του προγράμματος από την διεύθυνση 0843H.

5η Άσκηση

```

α)

MVI A, 0DH
SIM
LXI H, 00H
MVI C, 64d
EI
ADDR:
MVI A, C
CPI 00H

```

```

JNZ ADDR
DI
DAD H
DAD H
DAD H
MOV A, L
ANI 80H
MVI L, 00H
CPI 00H
JNZ ROUNDING
BACK:
HLT
ROUNDING:
INR H
JMP BACK
0034:
PUSH PSW
MOV A, C
ANI 00000001H
JPO 4MSB
IN 20H
ANI 00001111b
MOV B, A
JMP 4LSB
4MSB:
IN 20H
ANI 00001111b
RLC
RLC
RLC
RLC
ORA B
MVI D, 00H
MOV E, A
DAD D
4LSB:
POP PSW
DCR C
EI
RET

```

β)

```

LXI H, 00H
MVI C, 64d
MAIN:
IN 20H
ANI 80H
JP MAIN
MOV A, C
ANI 00000001b
JPO 4MSB
IN 20H
ANI 00001111b
MOV B, A
JMP 4LSB
4MSB:
IN 20H
ANI 00001111b
RLC
RLC
RLC
RLC
ORA B
MVI D, 00H

```

```
    MOV E, A
    DAD D
4LSB:
    DCR C
    JZ ADDR
CHECK:
    IN 20H
    ANI 80H
    JM CHECK
    JMP MAIN
ADDR:
    DAD H
    DAD H
    DAD H
    MOV A, L
    ANI 80H
    MVI L, 00H
    CPI 00H
    JNZ ROUNDING
BACK:
    HLT
ROUNDING:
    INR H
    JMP BACK
```

