

3^η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ

Αϊβαλής Θεόδωρος 03117099

Σαλιαράκης Πάυλος 03117135

1^η ΑΣΚΗΣΗ:

Στο Mlab θέλουμε να γραφεί πρόγραμμα Assembly, που να ελέγχει μέσω της διακοπής τύπου RST 6.5 τα φώτα ενός χώρου. Όταν προκαλείται διακοπή τύπου RST 6.5 να αναβοσβήνουν (με περίοδο ~1/2 sec) όλα τα LED της πόρτας εξόδου. Αυτό να παραμένει για περίπου ένα (1) λεπτό της ώρας και μετά να σβήνει. Αν όμως ενδιάμεσα ξαναενεργοποιηθεί η διακοπή να ανανεώνεται ο χρόνος του ενός λεπτού. Ο χρόνος που υπολείπεται να απεικονίζεται σε sec συνεχώς στα 2 δεξιάτερα δεκαεξαδικά ψηφία των 7-segments displays και σε δεκαδική μορφή.

```
START:
    IN 10H
    LXI H,0A00H          ; ΦΟΡΤΩΝΩ ΣΤΟΝ HL ΤΗΝ 1η ΘΕΣΗ ΤΗΣ ΜΝΗΜΗΣ
    MVI M,10H
    INX H                 ; ΔΕΙΧΝΩ ΣΤΗΝ ΕΠΟΜΕΝΗ ΘΕΣΗ 0A00H...0A05H
    MVI M,10H
    INX H
    MVI M,10H
    INX H
    MVI M,10H

    MVI A,0DH            ; 0DH = 00001101. ΦΤΙΑΧΝΩ ΤΗ ΜΑΣΚΑ ΠΟΥ ΕΠΙΤΡΕΠΕΙ
    TH ΔΙΑΚΟΠΗ
    SIM                  ; RST6.5 ΚΑΙ ΑΠΕΝΕΡΓΟΠΟΙΕΙ ΤΙΣΔΙΑΚΟΠΕΣ RST5.5 &
    7.5
    EI

WAIT:                    ; ΠΕΡΙΜΕΝΩ ΓΙΑ ΔΙΑΚΟΠΗ
    JMP WAIT

BCD:
    PUSH PSW             ; ΧΡΗΣΙΜΕΥΕΙ ΓΙΑ ΤΟ ΠΡΩΤΟ ΠΑΤΗΜΑ ΤΟΥ ΔΙΑΚΟΠΤΗ ΠΟΥ
    ΕΙΝΑΙ
    PUSH B               ; ΑΝΑΓΚΑΙΑ Η ΦΥΛΑΞΗ ΤΩΝ ΚΑΤΑΧΩΡΗΤΩΝ ΚΑΙ ΤΟΥ PSW
    PUSH D
    PUSH H
    MVI B,FFH            ; ΒΑΖΩ ΣΤΟ B ΤΟ FF(11111111)
    MOV A,E

DECA:
    INR B                ; ΑΥΞΑΝΩ ΤΟ B ΠΟΥ ΕΙΝΑΙ Ο ΜΕΤΡΗΤΗΣ ΤΩΝ ΔΕΚΑΔΩΝ
    SUI 0AH              ; ΑΦΑΙΡΩ ΑΠΟ ΤΟ A 10(ΔΕΚΑΔΙΚΟ), ΜΕΙΩΝΟΝΤΑΣ ΕΤΣΙ
    ΚΑΤΑ ΜΙΑ ΔΕΚΑΔΑ ΤΟΝ ΑΡΙΘΜΟ
    JNC DECA             ; ΣΥΝΕΧΙΖΩ ΝΑ ΑΦΑΙΡΩ ΑΝ Η ΣΗΜΑΙΑ CY ΔΕΝ ΕΙΝΑΙ
    ΜΗΔΕΝ

    ADI 0AH              ; ΠΡΟΣΘΕΤΩ 10 ΓΙΑ ΝΑ ΔΙΟΡΘΩΣΩ ΤΗ ΔΕΚΑΔΑ ΠΟΥ
    ΑΦΑΙΡΕΣΑ ΠΡΙΝ
    ; ΕΝΩ ΔΕΝ ΧΡΕΙΑΖΟΤΑΝ

    LXI H,0A04H
    MOV M,A              ; ΑΠΟΘΗΚΕΥΩ ΤΙΣ ΜΟΝΑΔΕΣ ΣΤΟ 5ο ΨΗΦΙΟ ΤΟΥ LCD

    INX H
    MOV M,B              ; ΑΠΟΘΗΚΕΥΩ ΤΙΣ ΜΟΝΑΔΕΣ ΣΤΟ 6ο ΨΗΦΙΟ ΤΟΥ LCD

    LXI D,0A04H          ; ΕΜΦΑΝΙΖΩ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΣΤΑ 2 ΔΕΞΙΟΤΕΡΑ ΨΗΦΙΑ
    CALL STDm            ; ΚΑΙ ΚΑΛΩ ΤΗΝ STDm

    LXI B,0032H          ; BC = 50 ---> DELB 50 msec delay
```

```

        MVI A,14H                ; A = 250 --> 50 msec X 20 = 1 sec delay
ONESECCLOOP:
        CALL DCD                ; ΕΚΤΥΠΩΝΩ
        CALL DELB               ; ΚΑΙ ΕΦΑΡΜΟΖΩ ΚΑΘΥΣΤΕΡΗΣΗ
        DCR A                   ; ΜΕΙΩΝΩ ΤΟ Α ΠΟΥ ΜΕΤΡΑΕΙ ΤΙΣ 20 ΕΠΑΝΑΛΗΨΕΙΣ
        CPI 00H                 ; ΑΝ ΔΕΝ ΕΙΝΑΙ ΜΗΔΕΝ ΣΥΝΕΧΙΖΩ
        JNZ ONESECCLOOP

        POP H                   ; ΑΠΟΚΑΤΑΣΤΑΣΗ ΤΩΝ ΚΑΤΑΧΩΡΗΤΩΝ
        POP D
        POP B
        POP PSW
        RET

INTR_ROUTINE:                   ; ΡΟΥΤΙΝΑ ΕΞΥΠΗΡΕΤΗΣΗΣ ΔΙΑΚΟΠΗΣ
        POP H                   ; ΚΑΝΩ pop PC ΑΠΟ ΤΗ ΣΤΟΙΒΑ

        LXI B,0064H
        MVI A,FFH               ; ΑΠΟΘΗΚΕΥΩ ΤΟ 11111111 ΓΙΑ ΝΑ ΑΝΑΨΟΥΝ ΟΛΑ ΤΑ
ΛΑΜΠΑΚΙΑ
        CMA                     ; ΑΝΤΙΣΤΡΕΦΩ ΕΠΕΙΔΗ ΕΙΝΑΙ ΑΡΝΗΤΙΚΗΣ ΛΟΓΙΚΗΣ
        CALL DELB               ; ΚΑΛΩ ΚΑΘΥΣΤΕΡΗΣΗ
        CALL DELB
        CALL DELB
        CALL DELB
        CALL DELB
        STA 3000H               ; ΕΜΦΑΝΙΖΩ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΣΤΑ ΛΑΜΠΑΚΙΑ

        MVI A,00H               ; ΕΔΩ ΕΚΑΝΑ ΑΛΛΑΓΗ ΓΙΑ ΝΑ ΑΝΑΒΟΣΒΗΝΕΙ
        CMA
        CALL DELB               ; ΚΑΛΩ ΚΑΘΥΣΤΕΡΗΣΗ
        CALL DELB
        CALL DELB
        CALL DELB
        CALL DELB
        STA 3000H

        MVI E,3CH
        EI                       ; ENABLE INTERRUPT

LOOP1:
        MVI A,FFH               ; ΑΠΟΘΗΚΕΥΩ ΤΟ 11111111 ΓΙΑ ΝΑ ΑΝΑΨΟΥΝ ΟΛΑ ΤΑ
ΛΑΜΠΑΚΙΑ
        CMA                     ; ΑΝΤΙΣΤΡΕΦΩ ΕΠΕΙΔΗ ΕΙΝΑΙ ΑΡΝΗΤΙΚΗΣ ΛΟΓΙΚΗΣ
        CALL DELB               ; ΕΦΑΡΜΟΖΩ ΚΑΘΥΣΤΕΡΗΣΗ
        CALL DELB
        CALL DELB
        CALL DELB
        CALL DELB
        CALL DELB
        STA 3000H               ; ΕΜΦΑΝΙΖΩ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΣΤΑ ΛΑΜΠΑΚΙΑ

        MVI A,00H               ; ΣΒΗΝΩ ΟΛΑ ΤΑ ΛΑΜΠΑΚΙΑ
        CMA
        CALL DELB               ; ΕΦΑΡΜΟΖΩ ΚΑΘΥΣΤΕΡΗΣΗ
        CALL DELB
        CALL DELB
        CALL DELB
        CALL DELB
        STA 3000H               ; ΕΜΦΑΝΙΖΩ ΑΠΟΤΕΛΕΣΜΑ
        CALL BCD                 ; ΚΑΛΩ ΤΗΝ BCD ΓΙΑ ΝΑ ΒΡΩ ΤΗΝ ΔΥΑΔΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ
ΚΑΤΟ ΠΑΡΟΥΣΙΑΖΩ
        DCR E                   ; ΜΕΙΩΝΩ ΤΟΝ ΜΕΤΡΗΤΗ ΤΩΝ ΕΠΑΝΑΛΗΨΕΩΝ Ε
        MOV A,E
        CPI 00H                 ; ΕΠΑΝΑΛΑΜΒΑΝΩ ΜΕΧΡΙ ΝΑ ΜΗΔΕΝΙΣΕΙ Ο ΜΕΤΡΗΤΗΣ
        JNZ LOOP1               ; ΓΙΑ ΤΑ 60 sec ΔΗΛΑΔΗ

```

```

MVI A,00H          ; ΜΕΤΑ ΤΑ 60 sec ΠΡΕΠΕΙ ΝΑ ΤΑ ΣΒΗΣΩ ΟΛΑ
CMA                ; ΚΑΝΩ ΤΗΝ ΑΝΤΙΣΤΡΟΦΗ
STA 3000H          ; ΕΜΦΑΝΙΖΩ ΤΟ ΑΠΟΤΕΛΕΣΜΑ 00000000

JMP WAIT          ; ΠΕΡΙΜΕΝΕΙ ΤΗΝ ΕΠΟΜΕΝΗ ΔΙΑΚΟΠΗ ΜΕ ΤΑ ΛΑΜΠΑΚΙΑ
ΣΒΗΣΤΑ
END

```

2^η ΑΣΚΗΣΗ :

Θέλουμε να υλοποιηθεί και να εκτελεστεί στο μLAB πρόγραμμα σε assembly που όταν προκαλείται διακοπή τύπου RST 6.5 να διαβάζει τα 2 διαδοχικά δεκαεξαδικά ψηφία ενός αριθμού (00-FFH => 0-255) που δίνονται στη συνέχεια από το πληκτρολόγιο και να τα απεικονίζει στα 2 αριστερότερα 7-segment display . Να συγκρίνει την τιμή αυτή με τρία κατώφλια K1, K2 και K3 με $K1 < K2 < K3$, που οι τιμές τους βρίσκονται στους καταχωρητές C, D και E αντίστοιχα. Στην συνέχεια να ανάβει ένα από τα τέσσερα LSB LED εξόδου που αντιστοιχούν στις περιοχές τιμών [0..K1], (K1..K2], (K2..K3] και (K3..FFH].

```

IN 10H              ;ΑΡΣΗ ΤΗΣ ΠΡΟΣΤΑΣΙΑΣ ΜΝΗΜΗΣ
START:
LXI H,0A00H         ;ΞΕΚΙΝΑΜΕ ΑΠΟ ΤΗ ΘΕΣΗ ΜΝΗΜΗΣ 0A00 ΓΙΑ ΤΗΝ ΕΚΤΥΠΩΣΗ ΣΤΟ
DISPLAY
MVI M,10H           ;ΤΟ 10H ΣΗΜΑΙΝΕΙ ΟΤΙ ΔΕΝ ΕΚΤΥΠΩΝΟΥΜΕ ΤΙΠΟΤΑ ΣΤΟ
ΑΝΤΙΣΤΟΙΧΟ SEVEN-SEGMENT LED
INX H
MVI M,10H
INX H
MVI M,10H
INX H
MVI M,10H           ;ΚΑΝΟΥΜΕ ΤΗ ΔΙΑΔΙΚΑΣΙΑ ΑΥΤΗ ΓΙΑ ΤΙΣ ΤΕΣΣΕΡΙΣ ΘΕΣΕΙΣ
ΜΝΗΜΗΣ 0A00H-0A03H ΠΟΥ ΤΙΣ ΑΝΤΙΣΤΟΙΧΙΖΟΥΜΕ ΣΤΑ ΤΕΣΣΕΡΑ ΔΕΞΙΟΤΕΡΑ ΨΗΦΙΑ ΤΟΥ
DISPLAY

```

```

MVI C,0AH           ;ΤΟ ΠΡΩΤΟ ΟΡΙΟ K1 ΙΣΟΥΤΑΙ ΜΕ 10 ΚΑΙ ΤΟΠΟΘΕΤΕΙΤΑΙ ΣΤΟΝ
ΚΑΤΑΧΩΡΗΤΗ C
MVI D,64H           ;ΤΟ ΔΕΥΤΕΡΟ K2 ΙΣΟΥΤΑΙ ΜΕ 100 ΚΑΙ ΤΟΠΟΘΕΤΕΙΤΑΙ ΣΤΟΝ
ΚΑΤΑΧΩΡΗΤΗ D
MVI E,C8H           ;ΤΟ ΤΡΙΤΟ K3 ΙΣΟΥΤΑΙ ΜΕ 200 ΚΑΙ ΤΟΠΟΘΕΤΕΙΤΑΙ ΣΤΟΝ
ΚΑΤΑΧΩΡΗΤΗ E
INR C               ;ΑΥΞΑΝΟΥΜΕ ΚΑΤΑ 1 ΤΑ ΟΡΙΑ ΓΙΑ ΜΕΤΕΠΕΙΤΑ ΕΛΕΓΧΟΥΣ
INR D
INR E

```

```

MVI A,0DH           ;ΚΑΤΑΧΩΡΟΥΜΕ ΣΤΟΝ A ΤΗΝ ΤΙΜΗ 00001101 ΠΡΟΚΕΙΜΕΝΟΥ ΝΑ
ΕΝΕΡΓΟΠΟΙΗΣΟΥΜΕ ΜΟΝΟ ΤΗ ΜΑΣΚΑ ΤΗΣ ΔΙΑΚΟΠΗΣ RST6.5
SIM
EI                  ;ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΩΝ ΔΙΑΚΟΠΩΝ

```

```

WAIT:
JMP WAIT           ;ΑΤΕΡΜΩΝ ΒΡΟΧΟΟΣ

```

```

INTR_ROUTINE:       ;ΡΟΥΤΙΝΑ ΕΞΥΠΗΡΕΤΗΣΗΣ ΔΙΑΚΟΠΗΣ
MVI D,64H           ;ΕΠΑΝΑΡΧΙΚΟΠΟΙΟΥΜΕ ΤΟΥΣ ΚΑΤΑΧΩΡΗΤΕΣ D,E ΔΙΟΤΙ ΑΡΓΟΤΕΡΑ
ΧΑΝΟΥΝ ΤΙΣ ΤΙΜΕΣ ΤΩΝ ΟΡΙΩΝ K2 ΚΑΙ K3
MVI E,C8H
INR D
INR C

```

```

POP H ;ΑΠΟΜΑΚΡΥΝΣΗ ΑΠΟ ΤΗ ΣΤΟΙΒΑ ΤΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΤΟΥ PC
ΠΟΥ ΑΠΟΘΗΚΕΥΤΗΚΕ ΜΕ ΤΗ ΔΙΑΚΟΠΗ
CALL KIND ;ΔΙΑΒΑΖΟΥΜΕ ΑΠΟ ΤΟ ΠΛΗΚΤΡΟΛΟΓΙΟ
LXI H,0A04H ;ΔΙΑΒΑΖΟΥΜΕ ΤΟ ΠΡΩΤΟ ΔΕΚΑΕΞΑΔΙΚΟ ΨΗΦΙΟ ΠΟΤ ΑΝΤΙΣΤΟΙΧΕΙ
ΣΤΑ 4 LSBs ΚΑΙ ΤΟ ΒΑΖΟΥΜΕ ΣΤΗ ΘΕΣΗ ΜΝΗΜΗΣ 0A04
MOV M,A
INX H
CALL KIND ;ΔΙΑΒΑΖΟΥΜΕ ΤΟ ΔΕΥΤΕΡΟ ΔΕΚΑΕΞΑΔΙΚΟ ΨΗΦΙΟ ΠΟΥ
ΑΝΤΙΣΤΟΙΧΕΙ ΣΤΑ 4 MSBs
MOV M,A ; ΤΟ ΤΟΠΟΘΕΤΟΥΜΕ ΣΤΗ ΘΕΣΗ ΜΝΗΜΗΣ 0A05 ΠΟΥ
ΑΝΤΙΣΤΟΙΧΙΖΕΤΑΙ ΣΤΟ ΑΡΙΣΤΕΡΟ LED ΤΟΥ DISPLAY
RLC
RLC
RLC
RLC ;ΚΑΝΟΥΜΕ 4 ΑΡΙΣΤΕΡΕΣ ΠΕΡΙΣΤΡΟΦΕΣ ΓΙΑ ΝΑ ΦΕΡΟΥΜΕ ΤΑ
BITS ΣΤΙΣ ΚΑΤΑΛΛΗΛΕΣ ΘΕΣΕΙΣ
LXI H,0A04H
MOV B,M
ADD B
MOV B,A ; Ο ΚΑΤΑΧΩΡΗΤΗΣ Β ΕΧΕΙ ΤΟ ΣΥΝΟΛΙΚΟ ΑΡΙΘΜΟ

MOV A,B
CMP C ;ΣΥΓΚΡΙΣΗ ΤΟΥ ΑΡΙΘΜΟΥ ΜΑΣ ΜΕ ΤΟ ΠΡΩΤΟ ΟΡΙΟ
JC FIRST_LSB ;ΑΝ ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟ Η ΙΣΟ ΤΟΤΕ ΘΑ ΑΝΑΨΕΙ ΤΟ ΠΡΩΤΟ LED
(ΑΠΟ ΔΕΞΙΑ)
CMP D ;ΣΥΓΚΡΙΣΗ ΜΕ ΤΟ ΔΕΥΤΕΡΟ ΟΡΙΟ
JC SECOND_LSB ;ΑΝ ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟ Η ΙΣΟ ΤΟΤΕ ΘΑ ΑΝΑΨΕΙ ΤΟ ΔΕΥΤΕΡΟ
LED
CMP E ;ΣΥΓΚΡΙΣΗ ΜΕ ΤΟ ΤΡΙΤΟ ΟΡΙΟ
JC THIRD_LSB ;ΑΝ ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟ Η ΙΣΟ ΤΟΤΕ ΘΑ ΑΝΑΨΕΙ ΤΟ ΤΡΙΤΟ LED

MVI A,08H ;ΑΝ ΕΙΝΑΙ ΜΕΓΑΛΥΤΕΡΟ ΑΠΟ ΤΟ ΤΡΙΤΟ ΟΡΙΟ ΤΟΤΕ ΘΑ ΠΡΕΠΕΙ
ΝΑ ΑΝΑΨΕΙ ΤΟ ΤΕΤΑΡΤΟ LED
CMA ;ΚΑΝΟΥΜΕ ΤΗΝ ΑΝΤΙΣΤΡΟΦΗ ΓΙΑΤΙ ΤΑ LED ΕΙΝΑΙ ΑΡΝΗΤΙΚΗΣ
ΛΟΓΙΚΗΣ
STA 3000H
JMP RESULT

FIRST_LSB:
MVI A,01H ;ΒΑΖΟΥΜΕ ΣΤΟΝ Α ΤΟ 00000001 ΩΣΤΕ ΝΑ ΑΝΑΨΕΙ ΜΟΝΟ ΤΟ
ΠΡΩΤΟ LED
CMA
STA 3000H
JMP RESULT

SECOND_LSB:
MVI A,02H ;ΒΑΖΟΥΜΕ ΣΤΟΝ Α ΤΟ 00000010 ΩΣΤΕ ΝΑ ΑΝΑΨΕΙ ΜΟΝΟ ΤΟ
ΔΕΥΤΕΡΟ LED
CMA
STA 3000H
JMP RESULT

THIRD_LSB:
MVI A,04H ;ΒΑΖΟΥΜΕ ΣΤΟΝ Α ΤΟ 00000100 ΩΣΤΕ ΝΑ ΑΝΑΨΕΙ ΜΟΝΟ ΤΟ
ΤΡΙΤΟ LED
CMA
STA 3000H
JMP RESULT

RESULT:
LXI D,0A00H ;ΒΑΖΟΥΜΕ ΣΤΟΝ ΔΙΠΛΟ ΚΑΤΑΧΩΡΗΤΗ D-E ΤΗ ΔΙΕΥΘΥΝΣΗ 0A00H
(ΔΕΞΙΟΤΕΡΗ ΘΕΣΗ)

```

```

CALL STDM
EI ;ΞΑΝΑΕΝΕΡΓΟΠΟΙΟΥΜΕ ΤΙΣ ΔΙΑΚΟΠΕΣ ΩΣΤΕ ΝΑ ΠΕΡΙΜΕΝΟΥΜΕ ΚΑΙ
ΤΗΝ ΕΠΟΜΕΝΗ

PRINT:
CALL DCD ;ΕΚΤΥΠΩΝΟΥΜΕ ΣΤΗΝ ΟΘΟΝΗ ΤΑ ΔΥΟ ΔΕΚΑΕΞΑΔΙΚΑ ΨΗΦΙΑ ΠΟΥ
ΕΧΟΥΜΕ ΠΑΡΕΙ ΑΠΟ ΤΟ ΠΛΗΚΤΡΟΛΟΓΙΟ ΣΤΑ ΔΥΟ ΑΡΙΣΤΕΡΟΤΕΡΑ 7-SEGMENT LED
JMP PRINT ;ΠΕΡΙΜΕΝΟΥΜΕ ΜΕΧΡΙ ΤΗΝ ΕΠΟΜΕΝΗ ΔΙΑΚΟΠΗ

END

```

3^η ΑΣΚΗΣΗ :

α) Θέλουμε να δώσουμε τη μακροεντολή INR16 ADDR που να αυξάνει έναν αριθμό X των 16 bit αποθηκευμένο σε 2 διαδοχικές θέσεις στη μνήμη ως εξής: XLOW=(ADDR) , XHIGH=(ADDR+1). Το αποτέλεσμα θα επιστρέφεται στις ίδιες θέσεις.

```

INR16 MACRO ADDR
PUSH PSW
PUSH H ;ΑΠΟΘΗΚΕΥΟΥΜΕ ΣΤΗ ΣΤΟΙΒΑ ΤΙΣ ΤΙΜΕΣ ΤΩΝ
ΚΑΤΑΧΩΡΗΤΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ
LHLD ADDR ;ΦΟΡΤΩΝΟΥΜΕ ΣΤΟ ΔΙΠΛΟ ΚΑΤΑΧΩΡΗΤΗ H-L ΤΟΝ 16-
BIT ΑΡΙΘΜΟ ΠΟΥ ΕΙΝΑΙ ΑΠΟΘΗΚΕΥΜΕΝΟΣ ΣΤΙΣ ΘΕΣΕΙΣ ADDR ΚΑΙ ADDR+1
MOV A,L
CPI FFH ;ΕΛΕΓΧΟΣ ΓΙΑ ΤΟ ΑΝ ΤΑ ΤΕΣΣΕΡΑ LSB ΕΙΝΑΙ 1111
ΚΑΙ ΑΝ ΕΙΝΑΙ ΤΑ ΜΗΔΕΝΙΖΟΥΜΕ ΚΑΙ ΠΡΟΣΘΕΤΟΥΜΕ 1 ΣΤΟ 8ο BIT ΤΟΥ ΑΡΙΘΜΟΥ
JNZ LABEL_1
MVI L,00H ; ΜΗΔΕΝΙΣΜΟΣ 8-LSB
MOV A,H ;ΕΛΕΓΧΟΣ ΓΙΑ ΤΟ ΑΝ ΤΑ 8 MSB ΕΙΝΑΙ ΚΑΙ ΑΥΤΑ
ΙΣΑ ΜΕ 1 ΚΑΙ ΑΝ ΕΙΝΑΙ ΤΑ ΜΗΔΕΝΙΖΟΥΜΕ
CPI FFH
JZ LABEL_2
INR H ;ΑΛΛΙΩΣ ΑΥΞΑΝΟΥΜΕ ΤΟ Η ΚΑΤΑ 1 ΟΠΩΣ
ΠΡΟΑΝΑΦΕΡΘΗΚΕ
JMP END_LABEL
LABEL_2:
MVI H,00H ;ΜΗΔΕΝΙΣΜΟΣ 8-LSB
JMP END_LABEL
LABEL_1:
INR L ;ΑΝ ΔΕΝ ΙΣΧΥΕΙ ΚΑΤΙ ΑΠΟ ΤΑ ΠΑΡΑΠΑΝΩ ΑΠΛΑ
ΑΥΞΑΝΟΥΜΕ ΤΟΝ ΚΑΤΑΧΩΡΗΤΗ L ΜΕ ΤΑ 8-LSB
END_LABEL
SHLD ADDR
POP H
POP PSW ;ΒΓΑΖΟΥΜΕ ΑΠΟ ΤΗ ΣΤΟΙΒΑ Ο,ΤΙ ΕΙΧΑΜΕ
ΑΠΟΘΗΚΕΥΣΕΙ
ENDM

```

β) Θέλουμε να δώσουμε τη μακροεντολή FILL ADDR, K, η οποία γεμίζει ένα τμήμα μνήμης με αρχική διεύθυνση ADDR και μήκος K με τους αριθμούς K, K-1, ..2, 1. Το μέγεθος του τμήματος K μπορεί να είναι από 0 έως 255. Για K=0 το μέγεθος του τμήματος να είναι ίσο με 256 και οι αριθμοί που θα αποθηκευθούν να είναι 0, 255, 254, ...1.

```

FILL MACRO ADDR,K
PUSH PSW
PUSH H

```

```

PUSH B ;ΑΠΟΘΗΚΕΥΟΥΜΕ ΣΤΗ ΣΤΟΙΒΑ ΤΙΣ ΤΙΜΕΣ ΤΩΝ ΚΑΤΑΧΩΡΗΤΩΝ
ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ
LXI H,ADDR ;ΒΑΖΟΥΜΕ ΤΗ ΔΙΕΥΘΥΝΣΗ ADDR ΣΤΟΝ ΔΙΠΛΟ ΚΑΤΑΧΩΡΗΤΗ
H-L
MVI B,K ;ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΤΟΝ Β ΩΣ ΚΑΤΑΧΩΡΗΤΗ
MVI M,K ;ΒΑΖΟΥΜΕ ΣΤΗ ΘΕΣΗ ΜΝΗΜΗΣ ADDR ΤΟ Κ
INX H ;ΑΥΞΑΝΟΥΜΕ ΤΗ ΘΕΣΗ ΜΝΗΜΗΣ

MVI A,K
CPI 00H ;ΕΛΕΓΧΟΣ ΓΙΑ ΤΟ ΑΝ ΤΟ Κ ΕΙΝΑΙ ΙΣΟ ΜΕ 0
JZ LABEL_1
LABEL_3:
DCR B ;ΜΕΙΩΝΟΥΜΕ ΤΟ ΜΕΤΡΗΤΗ
MOV M,B ;ΑΠΟΘΗΚΕΥΟΥΜΕ ΣΤΗΝ ΕΠΟΜΕΝΗ ΘΕΣΗ ΜΝΗΜΗΣ ΤΟΝ ΕΠΟΜΕΝΟ
ΑΡΙΘΜΟ
INX H ;ΑΥΞΑΝΟΥΜΕ ΤΗ ΔΙΕΥΘΥΝΣΗ
MOV A,B
CPI 00H ;ΕΛΕΓΧΟΥΜΕ ΑΝ ΕΧΟΥΜΕ ΚΑΛΥΨΕΙ ΟΛΕΣ ΤΙΣ ΘΕΣΕΙΣ
JZ LABEL_2 ;ΑΝ ΝΑΙ ΠΗΓΑΙΝΟΥΜΕ ΣΤΟ ΤΕΛΟΣ
JMP LABEL_3 ;ΑΝ ΟΧΙ ΕΠΑΝΑΛΑΜΒΑΝΟΥΜΕ ΤΗ ΔΙΑΔΙΚΑΣΙΑ

LABEL_1:
MVI B,FFH ;ΒΑΖΟΥΜΕ ΣΤΟΝ Β ΤΟ 255 ΓΙΑ ΤΗΝ ΠΕΡΙΠΤΩΣΗ ΠΟΥ ΤΟ Κ
ΕΙΝΑΙ ΙΣΟ ΜΕ 0 ΚΑΙ ΕΚΤΕΛΟΥΜΕ ΕΠΑΝΑΛΗΠΤΙΚΑ
JMP LABEL_3 ;ΤΗΝ ΠΑΡΑΠΑΝΩ ΔΙΑΔΙΚΑΣΙΑ

LABEL_2:
POP B
POP H
POP PSW ;ΕΠΑΝΑΦΕΡΟΥΜΕ ΣΤΟΥΣ ΚΑΤΑΧΩΡΗΤΕΣ ΤΙΣ ΑΡΧΙΚΕΣ ΤΟΥΣ
ΤΙΜΕΣ
ENDM

```

γ) Θέλουμε να δώσουμε τη μακροεντολή RHLR Q, R που περιστρέφει τα περιεχόμενα του κρατουμένου CY, των καταχωρητών Q και R κατά μια θέση αριστερά. Κάνουμε χρήση της στοίβας για την αποθήκευση και επαναφορά τιμής καταχωρητών.

```

RLHR MACRO Q,R
PUSH PSW ;ΑΠΟΘΗΚΕΥΟΥΜΕ ΣΤΗ ΣΤΟΙΒΑ ΤΙΣ ΤΙΜΕΣ ΤΩΝ ΚΑΤΑΧΩΡΗΤΩΝ
MOV A,R
RAL ;ΟΛΙΣΘΑΙΝΟΥΜΕ ΜΙΑ ΦΟΡΑ ΑΡΙΣΤΕΡΑ ΤΟΝ R ΚΑΝΟΝΤΑΣ
R0<-CY, CY<-R7
MOV R,A
MOV A,Q
RAL ;ΟΛΙΣΘΑΙΝΟΥΜΕ ΜΙΑ ΦΟΡΑ ΑΡΙΣΤΕΡΑ ΤΟΝ Q ΚΑΝΟΝΤΑΣ
Q0<-CY<-R7, CY<-Q7
MOV Q,A
POP PSW ;ΕΠΑΝΑΦΕΡΟΥΜΕ ΣΤΟΥΣ ΚΑΤΑΧΩΡΗΤΕΣ ΤΙΣ ΑΡΧΙΚΕΣ ΤΟΥΣ
ΤΙΜΕΣ
ENDM

```

4^η ΑΣΚΗΣΗ :

Στο μΕ 8085 εκτελείται η εντολή JMP 0900H. Ο μετρητής προγράμματος είναι (PC)=0800H και ο δείκτης σωρού (SP)=1FF0H. Στο μέσον της εκτέλεσης της εντολής συμβαίνει διακοπή RST 6.5. Δώστε τις νέες τιμές των PC, SP, το περιεχόμενο του σωρού καθώς και τις λειτουργίες που συμβαίνουν.

Αν η εκτέλεση της εντολής έχει ξεκινήσει τότε εκτελείται κανονικά και μετά την ολοκλήρωση εξυπηρετείται η διακοπή. Στη φάση της εξυπηρέτησης κρατείται ο PC ώστε να μπορεί να επιστρέψει ομαλά το πρόγραμμα στην κανονική ροή

Αρχικά ο μετρητής προγράμματος είναι (PC)=0800H και ο δείκτης στοίβας (SP)=1FF0H και εκείνη την στιγμή καλείται η ρουτίνα JMP 0900H. Μετά την ολοκλήρωση της JMP 0900H κρατάει τη διεύθυνση 0900H+1=0901H.

(SP) - 1 = PCH

(SP) - 2 = PCL

Η στοίβα μεγαλώνει από τις υψηλότερες προς τις χαμηλότερες διευθύνσεις. Άρα ο SP γίνεται 1FEEH.

Έπειτα γίνεται η διακοπή που στέλνει την εκτέλεση του προγράμματος στην θέση 0034H(διεύθυνση διακοπής). Μετά την ολοκλήρωση της ρουτίνας διακοπής εκτελείται το RET και η εκτέλεση του προγράμματος συνεχίζεται από τη θέση 0901H. Η στοίβα κάνει POP τις 2 θέσεις μνήμης που είχαν κρατηθεί για τα PCL και PCH και ο δείκτης στοίβας επιστρέφει στην θέση 1FF0H

5^η ΑΣΚΗΣΗ:

Θέλουμε να γραφεί πρόγραμμα Assembly σε μΥ-Σ 8085 που να λαμβάνει 16 δεδομένα των 8 bit από μια συσκευή. Το καθένα μεταφέρεται σε 2 βήματα μέσω των (X0-X3) της θύρας PORT_IN (20H) ενώ τα υπόλοιπα MSbit της θύρας (X4-X7) δεν χρησιμοποιούνται. Η συσκευή για κάθε 4 bit που αποστέλλει, προκαλεί πριν διακοπή RST5.5. Θέλουμε και να υπολογιστεί ο μέσος όρος των 16 δεδομένων με ακρίβεια 8 bit.

```
START:
MVI A,0EH          ; ΑΡΧΙΚΟΠΟΙΗΣΗ ΜΑΣΚΑΣ ΔΙΑΚΟΠΩΝ
SIM
LXI H,0000H        ; ΜΗΔΕΝΙΣΜΟΣ ΣΥΣΣΟΡΕΥΤΗ
MVI B,20H          ; ΒΑΖΩ ΣΤΟ Β ΤΟΝ ΜΕΤΡΗΤΗ ΓΙΑ ΤΙΣ 32 (ΔΕΚΑΔΙΚΟ) ΣΥΝΟΛΙΚΑ
ΔΙΑΚΟΠΕΣ
MVI D,00H          ; ΑΡΧΙΚΟΠΟΙΗΣΗ ΤΟΥ D ΣΤΟ ΜΗΔΕΝ
EI                ; ΕΝΕΡΓΟΠΟΙΗΣΗ ΔΙΑΚΟΠΩΝ

ADDR:              ; ΒΡΟΧΟΣ ΑΝΑΜΟΝΗΣ ΜΕΧΡΙ ΝΑ ΔΙΑΒΑΣΤΟΥΝ ΟΛΑ ΤΑ ΔΕΔΟΜΕΝΑ
MOV A,B
CPI 00H           ; ΕΛΕΓΧΩ ΑΝ Ο Β ΕΧΕΙ ΤΗΝ ΤΙΜΗ 0

JNZ ADDR         ; ΑΝ ΔΕΝ ΕΙΝΑΙ ΜΗΔΕΝ ΣΥΝΕΧΙΖΩ ΝΑ ΔΙΑΒΑΖΩ ΔΕΔΟΜΕΝΑ
DI              ; ΑΠΕΝΕΡΓΟΠΟΙΩ ΤΙΣ ΔΙΑΚΟΠΕΣΑΦΟΥ ΕΧΕΙ ΤΕΛΕΙΩΣΕΙ
              ; Η ΑΝΑΓΝΩΣΗ ΚΑΙ ΤΩΝ 32 ΔΕΔΟΜΕΝΩΝ
DAD H           ; ΥΠΟΛΟΓΙΖΩ ΤΩΡΑ ΤΟΝ ΜΕΣΟ ΟΡΟ ΟΛΙΣΘΑΙΝΩΝΤΑΣ 4
DAD H           ; ΘΕΣΕΙΣ ΠΡΟΣ ΤΑ ΑΡΙΣΤΕΡΑ ΤΟΝ ΔΙΠΛΟ ΚΑΤΑΧΩΡΗΤΗ H-L
DAD H
DAD H
HLT

RST5.5:
PUSH PSW
IN 20H
ANI 0FH          ; ΚΡΑΤΑΩ ΜΟΝΟ ΤΑ 4 LSBs
MOV C,A
```

```

MOV A,B          ; ΕΛΕΓΧΩ ΤΟΝ ΜΕΤΡΗΤΗ
RAR              ; ΑΝ ΕΙΝΑΙ ΑΡΤΙΟΣ Ή ΠΕΡΙΤΤΟΣ

JNC FIRST_FOUR   ; ΑΝ CY=0 ΤΟΤΕ ΕΙΝΑΙ ΑΡΤΙΟΣ ΚΑΙ ΚΑΝΩ JUMP ΣΤΗΝ
FIRST_FOUR
MOV A,C
ADD E
MOV E,A
DAD D            ; ΠΡΟΣΘΕΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ
                ; (H) (L) <- (H) (L) + (D) (E)

STEP:
DCR B            ; ΕΛΛΑΤΩΣΗ ΤΟΥ ΜΕΤΡΗΤΗ
POP PSW
EI
RET

FIRST_FOUR:
MOV A,C
RLC              ; ΚΑΝΩ ΑΡΙΣΤΕΡΗ ΟΛΙΣΘΗΣΗ ΚΑΙ ΠΑΩ ΤΑ 4 BIT ΠΟΥ ΑΠΟΜΟΝΩΣΑ
STA 4 MSB
RLC
RLC
RLC
MOV E,A          ; ΑΠΟΘΗΚΕΥΩ ΤΑ 4 LSBs ΣΤΟΝ E
JMP STEP

END

```