Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΑΚΑΔ. ΕΤΟΣ 2019-2020

Πεγειώτη Νάταλυ
03117707

3η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

Ασκήσεις Προσομοίωσης

1η ΑΣΚΗΣΗ

Κώδικας σε assembly

```
                IN 10H          ; Άρση προστασίας της μνήμης
                MVI A,0DH       ; Μάσκα για επίτρεψη της διακοπής RST6.5
                SIM             ; Εφαρμογή της μάσκας
                EI              ; Επίτρεψη διακοπών

WAIT:           JMP WAIT        ; Άλμα στην ετικέτα WAIT (επανάληψη μέχρι να γίνει διακοπή)

INTR_ROUTINE:   EI              ; Επίτρεψη διακοπών
                MVI A,10H       ; Αποθήκευση της τιμής 10H (κενό) στον A
                STA 0B02H       ; Φόρτωση της τιμής του A στις θέσεις μνήμης 0B05H – 0B02H
                STA 0B03H
                STA 0B04H
                STA 0B05H
                LXI B,01F4H     ; Καταχώρηση BC = 500 (για την καθυστέρηση)

INITDEC:        MVI A,06H       ; Φόρτωση της τιμής 06H (#6) στον A (δεκάδες)
                STA 0B01H       ; Αποθήκευση της τιμής του A στη θέση μνήμης 0B01H

INITMON:        MVI A,0AH       ; Φόρτωση της τιμής 0AH (#10) στον A (μονάδες)
                STA 0B00H       ; Αποθήκευση της τιμής του A στη θέση μνήμης 0B00H

DEC:            LDA 0B01H       ; Φόρτωση του περιεχομένου της διεύθυνσης 0B01H (δεκάδες) στον A
                DCR A           ; Μείωση του A κατά 1
                STA 0B01H       ; Αποθήκευση της τιμής του A στη θέση μνήμης 0B01H

MON:            LDA 0B00H       ; Φόρτωση του περιεχομένου της διεύθυνσης 0B00H (μονάδες) στον A
                DCR A           ; Μείωση του A κατά 1
                STA 0B00H       ; Αποθήκευση της τιμής του A στη θέση μνήμης 0B00H

                LXI D,0B00H ; Φόρτωση στον καταχωρητή D της διεύθυνσης της πρώτης θέσης
                              μνήμης που χρησιμοποιήθηκε για αποθήκευση (0B00H)
                CALL STDM   ; Αρχίζοντας από τη διεύθυνση που είναι αποθηκευμένη στο ζεύγος DE,
                              αποθήκευση των 6 πρώτων bytes στις θέσεις 0BF0H-0BF5H (RAM)
                CALL DCD    ; Εμφάνιση των bytes που βρίσκονται στη RAM (0BF0-0BF5) στο
                              display αρχίζοντας από δεξιά
                CALL DELB   ; Προσθήκη delay = BC*1ms = 500*10^-3 = 0,5s
```

```
LIGHT:      MVI A,00H   ; Φόρτωση της τιμής 00H (00000000) στον A
            STA 3000H   ; Ενημέρωση της κατάστασης των LED (on)
            CALL DELB   ; Προσθήκη delay = BC*1ms = 500*10^-3 = 0,5s

            MVI A,FFH   ; Φόρτωση της τιμής 00H (11111111) στον A
            STA 3000H   ; Ενημέρωση της κατάστασης των LED (off)
            CALL DELB   ; Προσθήκη delay = BC*1ms = 500*10^-3 = 0,5s

            LDA 0B00H   ; Φόρτωση του περιεχομένου της διεύθυνσης 0B00H (μονάδες) στον A
            CPI 00H     ; Σύγκριση του A με το 0, αν ισούνται Z = 1, αλλιώς Z = 0
            JNZ MON     ; Αν Z = 0, τότε άλμα στην ετικέτα MON (για συνέχεια μείωσης
                          μονάδων)
            LDA 0B01H   ; αλλιώς φόρτωση του περιεχομένου της διεύθυνσης 0B01H (δεκάδες)
                          στον A
            CPI 00H     ; Σύγκριση του A με το 0, αν ισούνται Z = 1, αλλιώς Z = 0
            JNZ INITMON ; Αν Z = 0, τότε άλμα στην ετικέτα INITMON (για μείωση δεκάδων)
            JMP WAIT    ; Άλμα στην ετικέτα WAIT (ολοκλήρωση της διαδικασίας)
END
```

2η ΑΣΚΗΣΗ

Κώδικας σε assembly

```
                IN 10H          ; Άρση προστασίας της μνήμης
                MVI A,0DH       ; Μάσκα για επίτρεψη της διακοπής RST6.5
                SIM             ; Εφαρμογή της μάσκας
                MVI A,10H       ; Αποθήκευση της τιμής 10H (κενό) στον A
                STA 0B03H       ; Φόρτωση της τιμής του A στις θέσεις μνήμης 0B03H – 0B00H
                STA 0B02H
                STA 0B01H
                STA 0B00H

MAIN:           EI              ; Επίτρεψη διακοπών
                JMP MAIN        ; Άλμα στην ετικέτα MAIN (επανάληψη μέχρι να γίνει διακοπή)

INTR_ROUTINE:   MVI A,10H       ; Αποθήκευση της τιμής 10H (κενό) στον A
                STA 0B05H       ; Φόρτωση της τιμής του A στις θέσεις μνήμης 0B05H – 0B04H
                STA 0B04H

                LXI D,0B00H ; Φόρτωση στον καταχωρητή D της διεύθυνσης της πρώτης
                            θέσης μνήμης που χρησιμοποιήθηκε για αποθήκευση (0B00H)
                CALL STDM ; Αρχίζοντας από τη διεύθυνση που είναι αποθηκευμένη στο ζεύγος DE,
                            αποθήκευση των 6 πρώτων bytes στις θέσεις 0BF0H-0BF5H (RAM)
                CALL DCD    ; Εμφάνιση των bytes που βρίσκονται στη RAM (0BF0-0BF5) στο
                            display αρχίζοντας από δεξιά
                MVI A,00H   ; Φόρτωση της τιμής 0 στον A
                CMA         ; Συμπλήρωμα ως προς 1 του A και καταχώρηση του στον ίδιο
                STA 3000H   ; Ενημέρωση της κατάστασης των LED (διεύθυνση 3000H)

                CALL KIND   ; Διάβασμα αριθμού από το πληκτρολόγιο και καταχώρηση του
                            αντίστοιχου κωδικού στον A
                STA 0B05H   ; Φόρτωση του περιεχομένου του A στη θέση μνήμης 0B05H
                CALL KIND   ; Διάβασμα αριθμού από το πληκτρολόγιο και καταχώρηση του
                            αντίστοιχου κωδικού στον A
                STA 0B04H   ; Φόρτωση του περιεχομένου του A στη θέση μνήμης 0B04H

                LXI D,0B00H ; Φόρτωση στον καταχωρητή D της διεύθυνσης της πρώτης
                            θέσης μνήμης που χρησιμοποιήθηκε για αποθήκευση (0B00H)
                CALL STDM ; Αρχίζοντας από τη διεύθυνση που είναι αποθηκευμένη στο ζεύγος DE,
                            αποθήκευση των 6 πρώτων bytes στις θέσεις 0BF0H-0BF5H (RAM)
                CALL DCD    ; Εμφάνιση των bytes που βρίσκονται στη RAM (0BF0-0BF5) στο
                            display αρχίζοντας από δεξιά
```

```
        MOV B,A     ; Φόρτωση του περιεχομένου του A στον B
        LDA 0B05H   ; Φόρτωση στον A του περιεχομένου της θέσης 0B05H
        RLC         ; Αριστερή περιστροφή των ψηφίων του A
        RLC         ; τέσσερις φορές, έτσι ώστε να μεταφερθούν
        RLC         ; στα 4 MSB του καταχωρητή
        RLC
        ORA B       ; Πράξη OR ανάμεσα στους καταχωρητές A και B ώστε να προκύψει σε
                    έναν καταχωρητή η τιμή που δημιουργείται από τα δύο ψηφία που
                    διαβάστηκαν από την οθόνη

        MVI C,3FH   ; Φόρτωση στον καταχωρητή C της τιμής 3FH (K1 = 63)
        MVI D,7FH   ; Φόρτωση στον καταχωρητή C της τιμής 7FH (K2 = 127)
        MVI E,BFH   ; Φόρτωση στον καταχωρητή C της τιμής BFH (K3 = 191)

        MVI H,01H   ; Φόρτωση στον καταχωρητή H της τιμής 01H (00000001)
                     για ενεργοποίηση του 1ου LED
        CMP C       ; Σύγκριση του περιεχομένου των A και C, αν ισούνται Z=1,
                    αλλιώς Z=0, αν A<C τότε CY = 1
        JC LIGHT    ; Αν CY = 1 τότε άλμα στην ετικέτα LIGHT
        JZ LIGHT    ; αλλιώς αν Z = 1, τότε άλμα στην ετικέτα LIGHT
        MVI H,02H   ; αλλιώς φόρτωση στον καταχωρητή H της τιμής 02H (00000010)
                     για ενεργοποίηση του 2ου LED
        CMP D       ; Σύγκριση του περιεχομένου των A και D, αν ισούνται Z=1,
                    αλλιώς Z=0, αν A<C τότε CY = 1
        JC LIGHT    ; Αν CY = 1 τότε άλμα στην ετικέτα LIGHT
        JZ LIGHT    ; αλλιώς αν Z = 1, τότε άλμα στην ετικέτα LIGHT
        MVI H,04H   ; αλλιώς φόρτωση στον καταχωρητή H της τιμής 04H (00000100)
                     για ενεργοποίηση του 3ου LED
        CMP E       ; Σύγκριση του περιεχομένου των A και E, αν ισούνται Z=1,
                    αλλιώς Z=0, αν A<C τότε CY = 1
        JC LIGHT    ; Αν CY = 1 τότε άλμα στην ετικέτα LIGHT
        JZ LIGHT    ; αλλιώς αν Z = 1, τότε άλμα στην ετικέτα LIGHT
        MVI H,08H   ; αλλιώς φόρτωση στον καταχωρητή H της τιμής 08H (00001000)
                     για ενεργοποίηση του 4ου LED

LIGHT:  MOV A,H     ; Φόρτωση του περιεχομένου του H στον A
        CMA         ; Συμπλήρωμα ως προς 1 του A και καταχώρηση του στον ίδιο
        STA 3000H   ; Ενημέρωση της κατάστασης των LED (διεύθυνση 3000H)
        LXI B,01F4H ; Προσθήκη καθυστέρησης
        CALL DELB
        JMP MAIN    ; Άλμα στην ετικέτα MAIN

END
```

Θεωρητικές Ασκήσεις

3η ΑΣΚΗΣΗ

α) INR16 MACRO ADDR

```
        PUSH PSW    ; Αποθήκευση των A και F στη στοίβα
        PUSH H      ; Αποθήκευση των H και L στη στοίβα
        LXI H,ADDR  ; Φόρτωση της διεύθυνσης ADDR στους H, L
        MVI A,01H   ; Φόρτωση της τιμής 1 στον A, έτσι ώστε να επιτευχθεί η αύξηση
        ADD M       ; Πρόσθεση του περιεχομένου της μνήμης HL στον A (A=A+M[HL])
        MOV M,A     ; Αποθήκευση του A (νέα αυξημένη τιμή) στη θέση μνήμης HL (ADDR)
        INX H       ; Αύξηση του  HL, δηλαδή μετάβαση στην επόμενη θέση μνήμης (ADDR+1)
        MVI A,00H   ; Φόρτωση της τιμής 0 στον A
        ADC M       ; Πρόσθεση της τιμής του A, του περιεχομένου της μνήμης θέσης μνήμης HL και
                    του κρατουμένου CY (μπορεί να προέκυψε από την αύξηση του περιεχομένου της
                    προηγούμενης θέσης μνήμης) (A=A+M[HL]+CY)
        MOV M,A     ; Αποθήκευση του A (πιθανή νέα αυξημένη τιμή) στη θέση μνήμης HL (ADDR+1)
        POP H       ; Επαναφορά των προηγούμενων τιμών των καταχωρητών από τη στοίβα
        POP PSW
    ENDM
```

β) FILL MACRO ADDR,K

```
            PUSH PSW        ; Αποθήκευση των A και F στη στοίβα
            PUSH H          ; Αποθήκευση των H και L στη στοίβα
            LXI H,ADDR      ; Φόρτωση της διεύθυνσης ADDR στους H, L
            MOV A,K         ; Φόρτωση της τιμής K στον A
            CPI 00H         ; Σύγκριση του A με το 0, αν ισούνται τότε Z = 1, αλλιώς Z=0
            JNZ NOT_ZERO    ; Αν Z=0, δηλαδή A≠0, τότε άλμα στην ετικέτα NOT_ZERO
FILL_MEM:   MVI M,00H       ; A=0, επομένως φόρτωση της τιμής 0 στην θέση μνήμης HL (ADDR)
            INX H           ; Αύξηση του HL, δηλαδή μετάβαση στην επόμενη θέση μνήμης
            MVI A,FFH       ; Αποθήκευση της τιμής 255 στον A
NOT_ZERO:   MOV M,A         ; Αποθήκευση του περιεχομένου του A στη θέση μνήμης HL
            INX H           ; Αύξηση του HL, δηλαδή μετάβαση στην επόμενη θέση μνήμης
            DCR A           ; Μείωση του περιεχομένου του A κατά 1
            CPI 00H         ; Σύγκριση του A με το 0, αν ισούνται τότε Z = 1, αλλιώς Z=0
            JNZ NOT_ZERO    ; Αν Z=0, δηλαδή A≠0, τότε άλμα στην ετικέτα NOT_ZERO

            POP H           ; Επαναφορά των προηγούμενων τιμών των καταχωρητών από τη
            POP PSW         στοίβα
ENDM
```

γ) RHLR MACRO Q,R
```
        MOV A,R     ; Αποθήκευση της τιμής R στον A
        RAL         ; Αριστερή περιστροφή του A και αποθήκευση του α7 στον CY
        MOV R,A     ; Αποθήκευση της νέας μετατοπισμένης τιμής του A στη μεταβλητή R
        MOV A,Q     ; Αποθήκευση της τιμής Q στον A
        RAL         ; Αριστερή περιστροφή του A και αποθήκευση του α7 στον CY
        MOV Q,A     ; Αποθήκευση της νέας μετατοπισμένης τιμής του A στη μεταβλητή Q
    ENDM
```

4η ΑΣΚΗΣΗ

Δεδομένα: (PC)=0800H
(SP)=1FF0H

Stack Pointer (SP)	Stack
1FF0H	……

Με την εκτέλεση της εντολής JMP 0900H, ο μετρητής προγράμματος παίρνει την τιμή (PC)=0900H.

Η διακοπή συμβαίνει στο μέσο της προηγούμενης εντολής, επομένως ολοκληρώνεται πρώτα αυτή (JMP 0900H) και μετά εκτελείται η διακοπή.
Η διακοπή RST6.5 σώζει το προηγούμενο PC (0900H) στη στοίβα, έτσι ώστε να μπορεί να επαναφέρει την τιμή αυτή και να επιστρέψει στο σημείο του προγράμματος όπου συνέβη η διακοπή.
Στη θέση (SP-1), δηλαδή 1FEFH, αποθηκεύεται η τιμή 09H και στην (SP-2), δηλαδή 1FEEH, αποθηκεύεται η τιμή 00H.

Stack Pointer (SP)	Stack
1FEEH	00H
1FEFH	09H
1FF0H	……

Ο SP γίνεται SP-2, λαμβάνει δηλαδή την τιμή (SP)=1FEEH.
Ο PC λαμβάνει τελικά την τιμή 0034H που είναι η διεύθυνση της διακοπής RST6.5.

Δηλαδή: (PC)=0034H
(SP)=1FEEH

5η ΑΣΚΗΣΗ

Κώδικας σε assembly

```
            MVI A,0EH   ; Αποθήκευση στον A της τιμής 0EH (επίτρεψη διακοπής RST5.5)
            SIM
            EI
            LXI H,0000H ; Μηδενισμός του ζεύγους καταχωρητών HL
            MVI B,20H   ; Αποθήκευση στον B (μετρητής) της τιμής 20H, δηλαδή 32

MAIN:       MOV A,B     ; Αποθήκευση στον A της τιμής του B
            CPI 00H     ; Σύγκριση του A με το 0, αν ισούνται τότε Z = 1, αλλιώς Z=0
            JNZ MAIN    ; Αν Z=0, δηλαδή A≠0, τότε άλμα στην ετικέτα MAIN
            DI          ; Απότρεψη των διακοπών, αφού πραγματοποιήθηκαν οι 32 επαναλήψεις
                        (A=0)
            DAD H       ; Υπολογισμός του μέσου όρου των 16 αριθμών με την εκτέλεση τεσσάρων
            DAD H       ; ολισθήσεων στα αριστερά (προσθέτοντας το HL στον εαυτό του)
            DAD H       ; ώστε το κλασματικό μέρος να αποθηκευτεί στον L
            DAD H
            HLT         ; Τέλος προγράμματος

RST5.5:     PUSH PSW    ; Αποθήκευση των A και F στη στοίβα
            IN 20H      ; Εισαγωγή δεδομένων (4 bits) στον A μέσω των X0-X3 της θύρας 20H
            MOV C,A     ; Αποθήκευση των δεδομένων του A στον C
            MOV A,B     ; Αποθήκευση της τιμής του B (μετρητής) στον A
            ANI 01H     ; Απομόνωση του LSB του A
            JNC EVEN    ; Αν A = 0 (ο μετρητής είναι άρτιος αριθμός → 4 MSB),
                        ; τότε άλμα στην ετικέτα EVEN
            MOV A,C     ; αλλιώς (4 LSB) επαναφορά των δεδομένων στον A
            ORA E       ; Αριθμητικό OR ανάμεσα στα ψηφία του A και του E
                        ; και αποθήκευση του αποτελέσματος στον A
            MOV E,A     ; Αποθήκευση του περιεχομένου του A (8bit  δεδομένα) στον E
            MVI D,00H   ; Αποθήκευση της τιμής 0 στον καταχωρητή D
            DAD D       ; Πρόσθεση περιεχομένου ζέυγους DE στο ζεύγος HL
            JMP EXIT    ; Άλμα στην ετικέτα EXIT

EVEN:       MOV A,C     ; Επαναφορά των δεδομένων στον A
            RRC         ; Δεξιά περιστροφή των ψηφίων του A
            RRC         ; τέσσερις φορές ώστε να μεταφερθούν τα δεδομένα
            RRC         ; στα 4 MSB του καταχωρητή
            RRC
            MOV E,A     ; Αποθήκευση του περιεχομένου του A στον E

EXIT:       DCR B       ; Μείωση του μετρητή B κατά 1
            POP PSW     ; Επαναφορά των προηγούμενων τιμών των καταχωρητών από τη στοίβα
            EI          ; Επίτρεψη διακοπών

            RET         ; Επιστροφή στη διεύθυνση από την οποία κλήθηκε η διακοπή
            END
```