ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ
ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 4

Ζαρίφης Νικόλαος ΑΜ 03112178
Ιωάννης Ορούτζογλου ΑΜ 03112124

Άσκηση 1:

```
IN 10H
MVI A,0DH
SIM
EI
JMP WAIT1
INTR_ROUTINE:
LXI B,03E8H ;1 s
DI ; close iderrupts
MVI A,00H ; Light all the led
STA 3000H
MVI C,3CH ; reset counter
EI ;open iderrupts
JMP WAIT2



WAIT1:
JMP WAIT1

WAIT2:
DCR C ; -1 second
JZ CLOSE
LXI H,0B04H ; place to store the numbers
CALL DELB
MOV A,C
JMP BCD1 ; BCD form of the number
KAT1:
ANI 0FH ;first 4 bytes to store
MOV M,A
MOV A,C
JMP BCD
KAT:
ANI F0H ;mask 4-MSB bits
RRC
RRC
RRC
RRC ; next 4 bits , the dozens
INX H
MOV M,A
LXI D,0B00H
;RST 1
CALL STDM
CALL DCD
JMP WAIT2

BCD:
PUSH D
;PUSH C
MVI E,00H ; dozens
MVI D,00H ; 0-9
```

```
FIND:
CPI 0AH ; 10 to find how many dozens
JC LAST
INR E ; +1 dozen

SUI 0AH
JMP FIND
LAST:
 ; A haze the other number in 0-4 bits.
MOV D,A ;
MOV A,E ;
RLC
RLC
RLC
RLC
ADD D ; now it has its BCD form
;POP C
POP D
JMP KAT

BCD1:
PUSH D
;PUSH C
MVI E,00H ; dozens
MVI D,00H ; 0-9
FIND2:
CPI 0AH ; 10 to find how many dozens
JC LAST2
INR E ; +1 dozen

SUI 0AH
JMP FIND2
LAST2:
 ; A haze the other number in 0-4 bits.
MOV D,A ;
MOV A,E ;
RLC
RLC
RLC
RLC
ADD D ; now it has its BCD form
;POP C
POP D
JMP KAT1

CLOSE:
MVI A,FFH
STA 3000H ; close leds
;POP PSW
;RET

JMP WAIT1
END
```

Άσκηση 2:

```
IN 10H
MVI C,F0H ; some numbers for testing
MVI B,09H ;
MVI A,0DH ;mask interupts
SIM
EI
```

```
JMP WAIT
INTR_ROUTINE:
DI ;disable interapts
CALL KIND
MOV H,A ; store input
CALL KIND
MOV L,A ; again
MOV A,L ;
STA 0B10H ; first digit for segment
MOV A,H ;
STA 0B11H ;
;RST 1
LXI D,0B10H ;
PUSH H
CALL STDM
CALL DCD ; let the segment light
POP H
MOV A,H ;
;RST 1
ANI 0FH
RLC
RLC
RLC
RLC
ADD L ; now A has the number that the segment display has
;RST 1
CMP B ; is A< = K1
JZ FIRSTLED
JC FIRSTLED
CMP C ; A < =K2
JZ SECONDLED
JC SECONDLED
MVI A,04H ;ELSE
CMA
STA 3000H
JMP ENDI
SECONDLED:
MVI A,02H ;
CMA ; second led
STA 3000H

JMP ENDI
FIRSTLED:
MVI A,01H
CMA
STA 3000H
ENDI:
EI ;enable interupts
RET

WAIT:
JMP WAIT

END
```

Άσκηση 3:

Την στιγμή που καλείτε η εντολή **CALL 3000H** , γίνεται το ακόλουθο:
((SP)-1) <- (PCH) =(20H)

((SP)-2) <- (PCL) = (00H)
(SP) <- 4000H-2 (= 3FFEH)
(PC) <- 3000H

Την στιγμή που γίνεται διακοπή τύπου RST 6.5:

((SP)-1) <- (PCH) =(30H)
((SP)-2) <- (PCL) =(00H)
(SP) <-(SP) -2 (=3FFCH)
(PC) <- 0034H (διεύθηνση διακοπής)

Και μόλις τελειώσει η ρουτίνα διακοπης:
(PCL) <-(SP) =(00H)
(PCH) <-((SP)+1) =(30H)
(SP) <-(SP) +2 =(3FFEH)

Και μόλις τελειώσει κι η ρούτηνα στην διεύθηνση : 3000H
(PCL) <-(SP) (=00H)
(PCH) <-((SP)+1) (=20H)
(SP) <-(SP) +2 =(4000H)

Άσκηση 4:
Θα χρησιμοποιήσουμε τους καταχωρήτες H-L για να αποθηκεύσουμε το άθροισμα των αριθμών. Δεν θα γίνει υπερχείλιση γιατί: 2^8 * 2^5 < 2^16 . Τέλος για να βρούμε το αποτέλεσμα αφού θέλουμε ακρίβεια 8 bit , επίσεις βλέπουμε ότι παντα τα 3 MSB του H θα είναι 0, άρα μπορούμε να κάνουμε όλισθηση 3 θέσεις χωρίς να χάσουμε δεδομένα. Θα το κανουμε αυτο προσθετοντας 2 φορες τον εαυτό του κι μετά χρησιμοποιώντας την ιδεα του macro της προηγουμενης σειράς ασκήσεων, θα μεταφερουμε όλα τα bit μια θέση αριστερά.Κι ο H , θα έχει το M.O., τέλος κάθε φόρα αλλάζω την διευθηνσή που θα κάνει JMP η διακοπή για να διαβάσω πρώτα τα LSB κι μετα τα MSB.

```
LXI H,FIRST
SHLD 0035H
MVI A,0DH
MVI C,20H ;COUNTER
MVI H,00H
MVI L,00H  ; prosthesi diplis akrivias
SIM ;mask for interupts
EI; ENABLE INTERUPTS
WAIT:
JMP WAIT

FIRST:
DI ;stop interupts
PUSH H
LXI H,SECOND ; for the next RST 5.5
```

```
SHLD 002DH
IN PORT_IN ; input
ANI 0FH ;mask number in case of rubbish
MOV B,A ; save value to B
POP H
EI ; enable inderupts
RET

SECOND :
DI ; same
PUSH H
LXI H,FIRST ; for the next interupt
SHLD 002DH
IN PORT_IN
RLC
RLC
RLC
RLC ;rotete to MSB
ANI F0H ; mask in case
ADD B ; now we have the number
POP H ; now we have the sum till now
MVI D,00H ;
MOV E,A ;
DAD D ; now we add the new value
DCR C ; -1 to counter
JZ ENDI
EI
RET



ENDI: ;find MEAN VALUE
DAD H
DAD H ; 4 thesis aristera , prepei alli mia thesi
MOV A,L
RLC ; now the carry has the MSB bit of L
MOV A,H
RAL ; now the MSB of L is the LSB of H
 ; Now H has the result
END
```