Ε. Μ. Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ & ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΑΘΗΝΑ 21. 9. 2021

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "***Συστήματα Μικροϋπολογιστών***"

(ΘΕΜΑ 3ο – ΣΥΝΟΛΟ 2 Μονάδες)

Έναρξη 13:40' - ΔΙΑΡΚΕΙΑ 30' + 10' Παράδοση: 14:20'

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Κυριακόπουλος Γεώργιος – el18153 . . . . . .

ΘΕΜΑ 3ο: (2 ΜΟΝΑΔΕΣ): Σε ένα προσωπικό υπολογιστή, να γραφεί πρόγραμμα σε Assembly μΕ 80x86 που να δέχεται από το πληκτρολόγιο τέσσερις (4) δεκαεξαδικούς αριθμούς (H_3, H_2, H_1, H_0 με τη σειρά αυτή) για να αποτελέσουν δυο μονοψήφιους και ένα διψήφιο δεκαεξαδικό αριθμό και να κάνει τον εξής υπολογισμό:
$P = H_3 + H_2 \times (H_1 \times 16 + H_0)$. Το πρόγραμμα τυπώνει στην οθόνη τα μηνύματα εισόδου και τους εισαγόμενους αριθμούς. Όταν συμπληρωθούν 4 έγκυροι δεκαεξαδικοί αριθμοί να αναμένει τον χαρακτήρα 'Η' και μετά να τυπώνει το αποτέλεσμα σε δεκαδική μορφή 3 ψηφίων αν δεν υπάρχει υπερχείλιση (δηλ. <1000). Αν όμως υπάρχει υπερχείλιση (>999) τότε να τυπώνει το μήνυμα OVERF, αυστηρά όπως φαίνεται παρακάτω:

DOSE 1ο ARITHMO = *D*
DOSE 2ο ARITHMO = *7*
DOSE 3ο ARITHMO = *5C*
APOTELESMA = *657* ή APOTELESMA = YPERX

Να θεωρήσετε δεδομένες τις μακροεντολές (σελ. 361-2, 373) του βιβλίου και μπορείτε να κάνετε χρήση των ρουτινών DEC_KEYB και PRINT_HEX χωρίς να συμπεριλάβετε τον κώδικά τους. Για την διευκόλυνσή σας, δίνονται οι πρώτες εντολές που αποτελούν τον 'σκελετό' του ζητούμενου προγράμματος.

ΑΠΑΝΤΗΣΗ

```
INCLUDE     MACROS

DATA_SEG    SEGMENT
  MSG1      DB 0AH,0DH, 'DOSE 1ο ARITHMO =  $'
  MSG2      DB 0AH,0DH, 'DOSE 2ο ARITHMO = $'
  MSG3      DB 0AH,0DH, 'DOSE 3ο ARITHMO = $'
  MSG4      DB 0AH,0DH, 'APOTELESMA = $'
DATA_SEG    ENDS

CODE_SEG    SEGMENT
  ASSUME   CS:CODE_SEG, DS:DATA_SEG

MAIN  PROC  FAR
      MOV   AX, DATA_SEG
      MOV   DS, AX
. . .
```

```
INCLUDE MACROS

DATA_SEG SEGMENT
    MSG1 DB 0AH,0DH, 'DOSE 1o ARITHMO = $'
    MSG2 DB 0AH,0DH, 'DOSE 2o ARITHMO = $'
    MSG3 DB 0AH,0DH, 'DOSE 3o ARITHMO = $'
    MSG4 DB 0AH,0DH, 'APOTELESMA = $'
    MSG5 DB 0AH,0DH, 'APOTELESMA = OVERF$'
DATA_SEG ENDS

CODE_SEG SEGMENT
    ASSUME CS:CODE_SEG, DS:DATA_SEG

MAIN PROC FAR
    MOV AX, DATA_SEG
    MOV DS, AX

START:
    PRINT_STR MSG1
    CALL HEX_KEYB
    CMP AL,'Q'
    JE QUIT

    MOV BL, AL          ; BL = H3

    PRINT_STR MSG2
    CALL HEX_KEYB
    CMP AL,'Q'
    JE QUIT

    MOV CL, AL          ; CL = H2

    PRINT_STR MSG3
    CALL HEX_KEYB
    CMP AL,'Q'
    JE QUIT

    AND AL,0FH
    ROL AL, 1
    ROL AL, 1
    ROL AL, 1
    ROL AL, 1

    MOV DL, AL          ; DL = H1*16

    CALL HEX_KEYB
    CMP AL,'Q'
    JE QUIT

    ADD AL, DL          ; AL = H1*16 + H0
    MOV AH, 0
    MUL CL              ; AX = H2*(H1*16 + H0)

    MOV BH, 0
    ADD AX, BX          ; AX = H3 + H2*(H1*16 + H0)
```

```
    MOV DX, AX

    WAIT_H:
        READ
        CMP AL, 'H'
        JNE WAIT_H

    CMP DX, 999
    JG OVERF

    MOV AX, DX
    MOV DL, AH

    CALL PRINT_HEX      ; MSB OF 3 DIGITS

    MOV DL, AL
    ROR DL, 1
    ROR DL, 1
    ROR DL, 1
    ROR DL, 1
    AND DL, 0FH
    CALL PRINT_HEX      ; 2ND DIGIT

    MOV DL, AL
    AND DL, 0FH
    CALL PRINT_HEX      ; 3RD DIGIT
    JMP START

    OVERF:
        PRINT_STR MSG5
        JMP START

    QUIT:
        EXIT
MAIN ENDP
```