

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 3<sup>ο</sup> – ΣΥΝΟΛΟ 2 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: . . . . .

**ΘΕΜΑ 3ο:** (2 ΜΟΝΑΔΕΣ): Σε ένα προσωπικό υπολογιστή, να γραφεί πρόγραμμα σε Assembly με 80x86 που να δέχεται από το πληκτρολόγιο τέσσερις (4) δεκαδικούς αριθμούς ( $D_3, D_2, D_1, D_0$  με τη σειρά αυτή) για να αποτελέσουν δυο μονοψήφιους και ένα διψήφιο δεκαδικό αριθμό και να κάνει τον εξής υπολογισμό:  $P = (D_3 + D_2) \times (D_1 \times 10 + D_0)$ . Το πρόγραμμα τυπώνει στην οθόνη τα μηνύματα εισόδου και τους εισαγόμενους αριθμούς. Όταν συμπληρωθούν 4 έγκυροι δεκαδικοί αριθμοί να αναμένει τον χαρακτήρα 'H' και μετά να τυπώνει το αποτέλεσμα σε δεκαεξαδική μορφή 3 ψηφίων αν είναι <500Hex, αλλιώς το μήνυμα overf, αυστηρά όπως φαίνεται παρακάτω:

GIVE 1ST NUMBER = 9  
GIVE 2ND NUMBER = 7  
GIVE 3ND NUMBER = 58  
RESULT = 3A0 ή RESULT = overf

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, 'GIVE 1ST NUMBER= \$'  
  MSG2       DB 0AH,0DH, 'GIVE 2ND NUMBER= \$'  
  MSG3       DB 0AH,0DH, 'GIVE 3ND NUMBER= \$'  
  MSG4       DB 0AH,0DH, 'RESULT = \$'  
  **MSG5       DB 0AH,0DH, 'overf'**  
DATA\_SEG    ENDS

CODE\_SEG    SEGMENT  
  ASSUME    CS:CODE\_SEG, DS:DATA\_SEG

MAIN PROC FAR  
  MOV   AX, DATA\_SEG  
  MOV   DS, AX  
ADDR1: PRINT\_STR MSG1  
      CALL DEC\_KEYB  
      PRINT AL  
      MOV BL, AL ; SAVE FIRST DIG  
      PRINT\_STR MSG2  
      CALL DEC\_KEYB  
      PRINT AL  
      ADD BL, D3 + D2  
      MOV BL, AL ; D3 + D2 to BL  
      PRINT\_STR MSG3  
      CALL DEC\_KEYB  
      PRINT AL  
      MOV CH, AL  
      CALL DEC\_KEYB

```

    PRINT AL
    MOV CL, AL ; NOW CX HAS CH FIRST PART CL SECOND PART
Await: MOV AL, 00H
    MOV AH, 8
    INT 21H
    PRINT AL
    CMP AL, 48H ; or CMP AL, 'H'
    JE CALC
    JMP Await
CALC: MOV DL, 10 ; allios kanoume MOV DL, 0AH
    MOV AL, CH
    MUL DL
    ADD CL ; AL HAS 10*D1 + D0
    MOV CH, 00H
    MOV CL, AL ; CL has 10*D1 + D0 or in hex 0A*D1 + D0
    MOV AH, 00H
    MOV AL, BL
    MUL CX ; MULTIPLY AX = (D3 + D2) with CX (D1*10 + D0)
    CMP AH, 05H
    JNB overflow_handle if AH >= 5 possible overflow (05 00 is 500 overflow)
CONTINUE:
    PRINT_STR MSG4
    PRINT AH
    PRINT AL
    RET
OVERFLOW_HANDLE:
    PRINT_STR MSG5
    RET

```