

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 3^ο – ΣΥΝΟΛΟ 2 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Θεodorής Παπαρηγόπουλος

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 'H' και μετά να τυπώνει το αποτέλεσμα σε δεκαδική μορφή 3 ψηφίων αν δεν υπάρχει υπερχειλίση (δηλ. <1000). Αν όμως υπάρχει υπερχειλίση (>999) τότε να τυπώνει το μήνυμα OVERF, αυστηρά όπως φαίνεται παρακάτω:

GIVE 1ST NUMBER = E
GIVE 2ND NUMBER = 7
GIVE 3ND NUMBER = 5D
RESULT = 665 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, 'RESULT = 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:

;; Proto minima gia H3
PRINT_STR MSG1
CALL HEX_KEYB
CMP AL, 'Q'
JE QUIT

MOV BL, AL

;; read H2
PRINT_STR MSG2

```
CALL HEX_KEYB
CMP AL,'Q'
JE QUIT
```

```
ADD AL, BL ;; BL = H3 + H2
MOV BL, AL
```

```
;; read H3
PRINT_STR MSG3
```

```
CALL HEX_KEYB ; read H1
CMP AL,'Q'
JE QUIT
```

```
MOV CL, 10H
MUL CL ;; AL = AL * CL → multiply by 16
MOV CL, AL ; copy to CL
```

```
; read H0
CALL HEX_KEYB
CMP AL,'Q'
JE QUIT
```

```
ADD AL, CL
MOV CL, AL ; 16*H1 + H0
```

```
MOV AL, BL
MUL CL ; AL = P = output
```

```
MOV CX,4 ;counter = 4 for proper prints
```

READ_CHAR:

```
CALL HEX_KEYB
CMP AL, 'H'
JE CHECK
JMP QUIT
```

CHECK:

```
CMP AL, 3E8H; 3E8 HEX = 1000
JG OVERFLOW ; JUMP LESS
JMP PRINT
```

PRINT:

```
ROL AX, 1 ; rotate AX once to get the LSB in the MSB position
ROL AX, 1
ROL AX, 1
ROL AX, 1
```

```
MOV DL, AL
AND DL, 0FH ; we keep only the 4 lsb of DL, that are the 4 MSB's result that get printed each time
PUSH AX
PRINT_STR MSG4
CALL PRINT_DEC
POP AX
LOOP PRINT
JMP START
```

OVERFLOW:

```
PRINT_STR MSG5
```

```
JMP START
QUIT:
MAIN ENDP
CODE_SEG ENDS
END MAIN
```