

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 3^ο – ΣΥΝΟΛΟ 2 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Θεodorής Παπαρηγόπουλος el18040

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 ένα διψήφιο και δυο μονοψήφιους δεκαδικούς αριθμούς και να κάνει τον εξής υπολογισμό: $P = (D_3 \times 10 + D_2) \times (D_1 + D_0)$. 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 <400Hex, αλλιώς το μήνυμα yperx, αυστηρά όπως φαίνεται παρακάτω:

DOSE 1ο ARITHMO = 58

DOSE 2ο ARITHMO = 7

DOSE 3ο ARITHMO = 9

APOTELESMA = 3A0 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 = \$'

MSG5 DB 0AH,0DH, APOTELESMA = yperx\$'

DATA_SEG ENDS

CODE_SEG SEGMENT

ASSUME CS:CODE_SEG, DS:DATA_SEG

MAIN PROC FAR

MOV AX, DATA_SEG

MOV DS, AX

MOV AX, DATA_SEG

MOV DS, AX

START:

PRINT_STR MSG1

CALL DEC_KEYB

CMP AL, 'Q'

JE QUIT

MOV BL, 10

MUL BL ; multiply first digit by 10

MOV BL, AL

```
CALL DEC_KEYB
CMP AL,'Q'
JE QUIT
ADD BL,AL ; add the number this way we create the first number in B
```

```
PRINT_STR MSG2
CALL DEC_KEYB
CMP AL,'Q'
JE QUIT
MOV DL,10
MUL DL
MOV DL,AL ; save the 3rd digit
```

```
PRINT_STR MSG3
CALL DEC_KEYB
CMP AL,'Q'
JE QUIT
ADD AL,DL ; add the 4th digit
MUL BL ; multiply with B
MOV CX,4 ;counter = 4 for proper prints
```

```
Read_char:
CALL DEC_KEYB ; read the h character
CMP AL, 'h'
JE CHECK
JMP QUIT
```

```
CHECK:
CMP 0400H ; check if we have overflow
JL PRINT
JMP OVERFLOW
```

```
PRINT:
PRINT_STR MSG4
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
CALL PRINT_HEX
POP AX
LOOP PRINT
JMP START
```

```
OVERFLOW:
PRINT_STR MSG5
JMP ADDR1
```

```
QUIT: EXIT
MAIN E
```