

Συστήματα Μικροϋπολογιστών 2018-2019

5η Ομάδα Ασκήσεων

Αρχικά δίνεται το περιεχόμενο του αρχείου μακροεντολών που χρησιμοποιήθηκε για τη μεταγλώττιση των προγραμμάτων. Για κάθε άσκηση της σειράς δίνεται μια περιγραφή της λειτουργίας του προγράμματος, ένα στιγμιότυπο (screenshot) της εικονικής οθόνης του προσομοιωτή *emu8086* με τις εξόδους του προγράμματος μετά από ικανό αριθμό εκτελέσεων και ο πλήρης κώδικας του προγράμματος.

Ως υλικό μελέτης και αναφοράς για τη συγγραφή του κώδικα των παρακάτω προγραμμάτων αξιοποιήθηκαν οι σχετικές διαφάνειες του μαθήματος για το έτος 2018-2019 (αρχεία [mP10_80x86.pdf](#) και [mP11_80x86_programs.pdf](#)) και διάφορες ιστοσελίδες του διαδικτύου.

Αρχείο μακροεντολών – *macros.asm*

```
;Εμφάνιση χαρακτήρα
PRINTCH MACRO CHAR
    PUSH AX
    PUSH DX
    MOV DL,CHAR
    MOV AH,2
    INT 21H
    POP DX
    POP AX
ENDM

;Εισαγωγή χαρακτήρα
READCH MACRO
    MOV AH,8
    INT 21H
ENDM

;Εισαγωγή και εμφάνιση χαρακτήρα
READNPRINTCH MACRO
    MOV AH,1
    INT 21H
ENDM

;Εμφάνιση string
PRINTSTR MACRO STRING
    PUSH AX
    PUSH DX
    MOV DX,OFFSET STRING
    MOV AH,9
    INT 21H
    POP DX
    POP AX
ENDM

;Εξόδος
EXIT MACRO
    MOV AX,4C00H
    INT 21H
ENDM

;Αλλαγή γραμμής
PRINTLN MACRO
    PUSH AX
    PUSH DX
    MOV DL,13
    MOV AH,2
    INT 21H
    MOV DL,10
    MOV AH,2
    INT 21H
    POP DX
    POP AX
ENDM

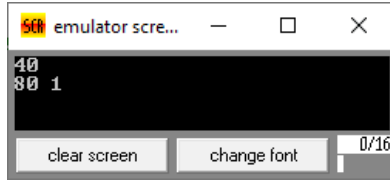
;Εσοχή
PRINTTAB MACRO
    PUSH AX
    PUSH DX
    MOV DL,9
    MOV AH,2
    INT 21H
    POP DX
    POP AX
ENDM
```

1η Άσκηση

Το πρόγραμμα αρχικά αποθηκεύει τους αριθμούς 128, 127, 126, ..., 3, 2, 1 σε 128 διαδοχικές 8-bit θέσεις μνήμης (δομή TABLE). Στη συνέχεια σαρώνει τον πίνακα TABLE για να βρει τους περιττούς αριθμούς (διαιρώντας τους με το 2 – μεταβλητή TWO), τους οποίους αθροίζει και μετρά και βρίσκει το μέγιστο και το ελάχιστο των αριθμών συγκρίνοντάς τους σειριακά με τον 1ο και τον τελευταίο αριθμό της σειράς αντίστοιχα. Τέλος, τυπώνει το ακέραιο μέρος του μέσου όρου (16 bit) και σε νέα γραμμή το μέγιστο και ελάχιστο με ένα κενό μεταξύ τους, όλα σε δεκαεξαδική μορφή.

Για την εκτύπωση των αριθμών στην οθόνη αξιοποιήθηκαν οι 2 ρουτίνες των σελίδων 17,18 των διαφανειών

mP11_80x86_programs.pdf: Η PRINT_HEX τυπώνει το δεκαεξαδικό ψηφίο που βρίσκεται στον DL και η PRINT_NUM8_HEX τυπώνει τον 8-bit αριθμό που βρίσκεται στον AL με χρήση της PRINT_HEX, με κατάλληλη τροποποίηση για να μην εμφανίζεται το αρχικό μηδενικό.



```
INCLUDE macros.asm
```

```
DATA SEGMENT
    TABLE DB 128 DUP(?)           ;Το σύνολο δεδομένων
    TWO DB DUP(2)                  ;Για τον έλεγχο της ισοτιμίας των αριθμών
DATA ENDS
```

```
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA

    MAIN PROC FAR
        MOV AX,DATA
        MOV DS,AX

        ;Αποθήκευση των αριθμών στη μνήμη
        MOV DI,0                    ;Δείκτης πίνακα αριθμών
        MOV CX,128                  ;Πλήθος αριθμών
    STORE:
        MOV TABLE[DI],CL
        INC DI
        LOOP STORE

        ;Έλεγχος ισοτιμίας, άθροιση και μέτρηση
        MOV DH,0                    ;Για την πρόσθεση AX+DL
        MOV AX,0                    ;Άθροισμα περιττών
        MOV BX,0                    ;Πλήθος περιττών
        MOV DI,0
        MOV CX,128
    FINDADDDODD:
        PUSH AX
        MOV AH,0                    ;Για τη διαίρεση AX/2
        MOV AL, TABLE[DI]          ;Έλεγχος ισοτιμίας
        DIV TWO                      ;
        CMP AH,0                    ;
        POP AX
        JE SKIPEVEN                  ;AX div 2 = 0 ?
        MOV DI, TABLE[DI]          ;Προσωρινή αποθήκευση
        ADD AX,DX                    ;Άθροιση
        INC BX                        ;Περιττός
    SKIPEVEN:
        ;Άρτιος
        INC DI
        LOOP FINDADDDODD

        ;Υπολογισμός μέσου όρου
        MOV DX,0                    ;Για τη διαίρεση AX/BX
        DIV BX                        ;Άθροισμα/πλήθος
        ;Εκτύπωση μέσου όρου
        CALL PRINT_NUM8_HEX ;Εκτύπωση του ακέραιου μέρους (πηλίκο)
        PRINTLN

        ;Εύρεση μέγιστου, ελάχιστου
        MOV AL, TABLE[0]            ;Αρχικό μέγιστο
        MOV BL, TABLE[127]          ;Αρχικό ελάχιστο
        MOV DI,0
        MOV CX,128
    MAXMIN:
        CMP AL, TABLE[DI]          ;Έλεγχος για μέγιστο
        JC NEWMAX
        JMP TOMIN
    NEWMAX:
        ;Νέο μέγιστο
        MOV AL, TABLE[DI]
```

```

    JMP NEXTNUM
TOMIN:                                ;Ελεγχος για ελάχιστο
    CMP TABLE[DI],BL
    JC  NEWMIN
    JMP NEXTNUM
NEWMIN:                                ;Νέο ελάχιστο
    MOV BL, TABLE[DI]
NEXTNUM:
    INC DI
    LOOP MAXMIN

                                ;Εκτύπωση μέγιστου, ελάχιστου
    CALL PRINT_NUM8_HEX ;Εκτύπωση μέγιστου
    PRINTCH ' '
    MOV AL,BL
    CALL PRINT_NUM8_HEX ;Εκτύπωση ελάχιστου

    EXIT
MAIN ENDP

                                ;Εκτύπωση 8-bit αριθμού σε δεκαεξαδική μορφή (από τον AL)
PRINT_NUM8_HEX PROC NEAR;βλ. mP11_80x86_programs.pdf σελ. 17
    MOV DL,AL
    AND DL,0F0H                ;1ο δεκαεξαδικό ψηφίο
    MOV CL,4
    ROR DL,CL
    CMP DL,0                    ;Αγνόηση αρχικού μηδενικού
    JE  SKIPZERO
    CALL PRINT_HEX
SKIPZERO:
    MOV DL,AL
    AND DL,0FH                ;2ο δεκαεξαδικό ψηφίο
    CALL PRINT_HEX
    RET
PRINT_NUM8_HEX ENDP

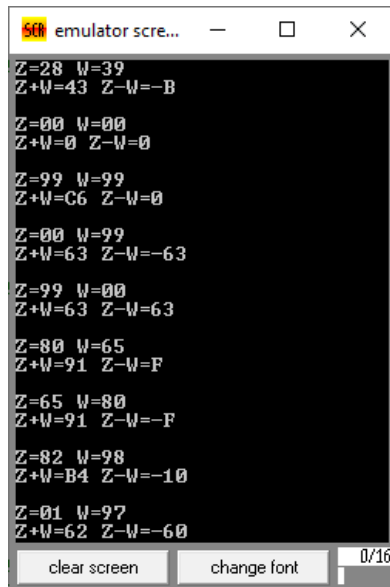
                                ;Εκτύπωση δεκαεξαδικού ψηφίου (από τον DL)
PRINT_HEX PROC NEAR ;βλ. mP11_80x86_programs.pdf σελ. 18
    CMP DL,9
                                ;0...9
    JG  LETTER
    ADD DL,48
    JMP SHOW
LETTER:
    ADD DL,55                ;A...F
SHOW:
    PRINTCH DL
    RET
PRINT_HEX ENDP
CODE ENDS
END MAIN

```

2η Άσκηση

Το πρόγραμμα δέχεται 2 διψήφιους δεκαδικούς αριθμούς, Z και W , από το πληκτρολόγιο και τους εμφανίζει στην οθόνη. Στη συνέχεια υπολογίζει το άθροισμα $Z+W$ και τη διαφορά τους $Z-W$ και τα εμφανίζει σε δεκαεξαδική μορφή στην επόμενη γραμμή. Το πρόγραμμα είναι συνεχούς λειτουργίας.

Οι 2 αριθμοί εισάγονται με κλήση της ρουτίνας `READ_DEC_DIGIT`, η οποία δέχεται δεκαδικά ψηφία και αφαιρεί το offset της κωδικοποίησης ASCII. Για κάθε αριθμό εισάγεται πρώτα το 1ο ψηφίο που πολλαπλασιάζεται με 10 (μεταβλητή `TEN`) για να προκύψουν οι δεκάδες και μετά το 2ο που είναι οι μονάδες και προστίθεται στο 1ο. Οι 2 αριθμοί αποθηκεύονται στη μνήμη ως bytes. Το άθροισμα υπολογίζεται με χρήση της εντολής `ADD` και η διαφορά με χρήση της εντολής `SUB`, αφού πρώτα γίνει έλεγχος για το ποιος αριθμός είναι μεγαλύτερος. Όταν ο 1ος αριθμός είναι μικρότερος απ' τον 2ο εμφανίζεται αρνητικό πρόσημο στη διαφορά. Η εμφάνιση των αποτελεσμάτων στην οθόνη γίνεται με κλήση της ρουτίνας `PRINT_NUM8_HEX` που τυπώνει 8-bit αριθμό που βρίσκεται στον `AL` με χρήση της `PRINT_HEX` που τυπώνει το δεκαεξαδικό ψηφίο που βρίσκεται στον `DL`, με κατάλληλη τροποποίηση για να μην εμφανίζονται αρχικά μηδενικά. Οι 2 τελευταίες ρουτίνες είναι παραλλαγές των αντίστοιχων ρουτινών των σελίδων 17,18 των διαφανειών *mP11_80x86_programs.pdf*.



```
INCLUDE macros.asm
```

```
DATA SEGMENT
```

```
MSGZ DB "Z=$"
MSGW DB "W=$"
MSGSUM DB "Z+W=$"
MSGSUB DB "Z-W=$"
MSGMINUS DB "Z-W=-$"
Z DB 0
W DB 0
TEN DB DUP(10)
```

```
;Για τις δεκάδες
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS:CODE, DS:DATA
```

```
MAIN PROC FAR
    MOV AX, DATA
    MOV DS, AX
```

```
START: ;Κατασκευή, εμφάνιση, αποθήκευση του Z
```

```
    PRINTSTR MSGZ
    CALL READ_DEC_DIGIT ;1ο ψηφίο (δεκάδες)
    MUL TEN
    LEA DI, Z ;Αποθήκευση 1ου ψηφίου
    MOV [DI], AL
    CALL READ_DEC_DIGIT ;2ο ψηφίο (μονάδες)
    ADD [DI], AL ;Αποθήκευση 2ου ψηφίου
```

```
    PRINTCH ' ' ;Κατασκευή, εμφάνιση, αποθήκευση του W
```

```
    PRINTSTR MSGW
    CALL READ_DEC_DIGIT ;1ο ψηφίο (δεκάδες)
    MUL TEN
    LEA DI, W ;Αποθήκευση 1ου ψηφίου
    MOV [DI], AL
    CALL READ_DEC_DIGIT ;2ο ψηφίο (μονάδες)
    ADD [DI], AL ;Αποθήκευση 2ου ψηφίου
```

```

PRINTLN
                                ;Αθροισμα
MOV AL,[DI]                    ;W
LEA DI,Z                      ;Z
ADD AL,[DI]                    ;Πρόσθεση
PRINTSTR MSGSUM
CALL PRINT_NUM8_HEX ;Εμφάνιση του αθροίσματος

PRINTCH ' '
                                ;Διαφορά
MOV AL,[DI]                    ;Z
LEA DI,W                      ;W
MOV BL,[DI]

CMP AL,BL                      ;Z>W ή W>Z ?
JB MINUS
SUB AL,BL                      ;Αφαίρεση για Z>W
PRINTSTR MSGSUB
JMP SHOWSUB
MINUS:
SUB BL,AL                      ;Αφαίρεση για Z<W
MOV AL,BL
PRINTSTR MSGMINUS
SHOWSUB:
CALL PRINT_NUM8_HEX ;Εμφάνιση της διαφοράς
PRINTLN
PRINTLN
JMP START
MAIN ENDP

                                ;Εισαγωγή και εμφάνιση δεκαδικού ψηφίου (στον AL)
READ_DEC_DIGIT PROC NEAR
READ:
    READCH
    CMP AL,48                  ;<0 ?
    JB READ
    CMP AL,57                  ;>9 ?
    JA READ
    PRINTCH AL
    SUB AL,48                  ;Κωδικός ASCII
    RET
READ_DEC_DIGIT ENDP

                                ;Εκτύπωση 8-bit αριθμού σε δεκαεξαδική μορφή (από τον AL)
PRINT_NUM8_HEX PROC NEAR;βλ. mP11_80x86_programs.pdf σελ. 17
    MOV DL,AL
    AND DL,0F0H                ;1ο δεκαεξαδικό ψηφίο
    MOV CL,4
    ROR DL,CL
    CMP DL,0                    ;Αγνόηση αρχικού μηδενικού
    JE SKIPZERO
    CALL PRINT_HEX
SKIPZERO:
    MOV DL,AL
    AND DL,0FH                  ;2ο δεκαεξαδικό ψηφίο
    CALL PRINT_HEX
    RET
PRINT_NUM8_HEX ENDP

                                ;Εκτύπωση δεκαεξαδικού ψηφίου (από τον DL)
PRINT_HEX PROC NEAR ;βλ. mP11_80x86_programs.pdf σελ. 18
    CMP DL,9                    ;0...9
    JG LETTER
    ADD DL,48
    JMP SHOW
LETTER:
    ADD DL,55                    ;A...F
SHOW:
    PRINTCH DL
    RET
PRINT_HEX ENDP
CODE ENDS
END MAIN

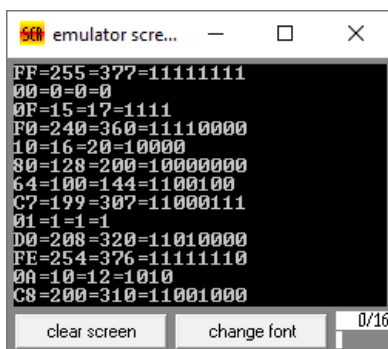
```

3η Άσκηση

Το πρόγραμμα δέχεται έναν 8-bit δεκαεξαδικό αριθμό από τον `BL` και τον εμφανίζει σε δεκαεξαδική, δεκαδική, οκταδική και δυαδική μορφή. Ανάμεσα στα αποτελέσματα τοποθετεί ένα `=`, ενώ αν δοθεί ο χαρακτήρας `T` τερματίζεται. Το πρόγραμμα είναι συνεχούς λειτουργίας.

Η εισαγωγή του αριθμού γίνεται καλώντας 2 φορές τη ρουτίνα `HEX_KEYB` των σχετικών διαφανειών (βλ. *mP11_80x86_programs.pdf* σελ. 20-21) που διαβάζει 1 δεκαεξαδικό ψηφίο, εκτός αν δοθεί `T` οπότε επιστρέφει αμέσως. Τα ψηφία “ενώνονται” και ο αριθμός αποθηκεύεται στον `BL`. Στη συνέχεια καλούνται διαδοχικά οι ρουτίνες `PRINT__DEC`, `PRINT__OCT` και `PRINT__BIN` που εμφανίζουν στην οθόνη έναν 8-bit αριθμό από τον `BL` σε δεκαδική, οκταδική και δυαδική μορφή αντίστοιχα. Τα ψηφία του αριθμού εξάγονται με διαδοχικές διαιρέσεις του με τη βάση του αριθμητικού συστήματος στο οποίο πρόκειται να αναπαρασταθεί (10 για δεκαδικό, 8 για οκταδικό, 2 για δυαδικό), μέχρι το πηλίκο των διαιρέσεων να μηδενιστεί και αποθηκεύονται προσωρινά στη στοίβα με σειρά από το LSB στο MSB. Μετά εξάγονται από τη στοίβα με αντίστροφη σειρά και τυπώνονται στην οθόνη, αφού προστεθεί σ’ αυτά το offset της κωδικοποίησης ASCII (βλ. *mP11_80x86_programs.pdf* σελ. 26-27).

Σημειώνεται ότι πιθανώς λόγω κάποιου bug του προσομοιωτή δεν ήταν δυνατό η ρουτίνα εκτύπωσης δεκαδικού να ονομαστεί `PRINT__DEC`, γι’ αυτό ονομάστηκε `PRINT__DEC`.



```
INCLUDE macros.asm
```

```
CODE SEGMENT
```

```
ASSUME CS:CODE
```

```
MAIN PROC FAR
```

```
START:
```

```
CALL HEX_KEYB      ;Εισαγωγή 1ου ψηφίου
CMP AL, 'T'         ;Ψηφίο = T ?
JE FINISH           ;Έλεγχος για τερματισμό
MOV BL, AL          ;Αποθήκευση 1ου ψηφίου
ROL BL, 4           ;
CALL HEX_KEYB      ;Εισαγωγή 2ου ψηφίου
CMP AL, 'T'
JE FINISH
OR BL, AL           ;Ένωση ψηφίων
```

```
PRINTCH '='
CALL PRINT__DEC     ;Εμφάνιση δεκαδικού
PRINTCH '='
CALL PRINT__OCT     ;Εμφάνιση οκταδικού
PRINTCH '='
CALL PRINT__BIN     ;Εμφάνιση δυαδικού
```

```
PRINTLN
JMP START
```

```
FINISH:
```

```
EXIT
```

```
MAIN ENDP
```

```
HEX_KEYB PROC NEAR      ;Εισαγωγή δεκαεξαδικού ψηφίου (στον AL)
                        ;βλ. mP11_80x86_programs.pdf σελ. 20-21
```

```
READ:
```

```
READCH
CMP AL, 'T'           ;=T ?
JE RETURN
CMP AL, 48             ;<0 ?
JL READ
CMP AL, 57             ;>9 ?
JG LETTER
PRINTCH AL
SUB AL, 48             ;Κωδικός ASCII
JMP RETURN
```

```
LETTER:               ;A...F
```

```
CMP AL, 'A'           ;<A ?
JL READ
```

```

    CMP AL,'F'                ;>F ?
    JG READ
    PRINTCH AL
    SUB AL,55                 ;Κωδικός ASCII
RETURN:

    RET
HEX_KEYB ENDP
                ;Εμφάνιση 8-bit δεκαδικού αριθμού (από τον BL)
PRINT__DEC PROC NEAR;βλ. mp11_80x86_programs.pdf σελ. 26-27
    PUSH BX

    MOV AL,BL
    MOV BL,10                 ;Δεκαδικός => διαιρέσεις με 10
    MOV CX,0                 ;Μετρητής ψηφίων
GETDEC:        ;Εξαγωγή ψηφίων
    MOV AH,0                 ;Αριθμός mod 10 (υπόλοιπο)
    DIV BL                   ;Διαίρεση με 10
    PUSH AX                 ;Προσωρινή αποθήκευση
    INC CL
    CMP AL,0                 ;Αριθμός div 10 = 0 ? (πηλίκο)
    JNE GETDEC
PRINTDEC:      ;Εμφάνιση ψηφίων
    POP AX
    ADD AH,48                 ;Κωδικός ASCII
    PRINTCH AH
    LOOP PRINTDEC

    POP BX
    RET
PRINT__DEC ENDP
                ;Εμφάνιση 8-bit οκταδικού αριθμού (από τον BL)
PRINT__OCT PROC NEAR
    PUSH BX

    MOV AL,BL
    MOV BL,8                 ;Οκταδικός => διαιρέσεις με 8
    MOV CX,0
GETOCT:
    MOV AH,0
    DIV BL
    PUSH AX
    INC CL
    CMP AL,0
    JNE GETOCT
PRINTOCT:
    POP AX
    ADD AH,48
    PRINTCH AH
    LOOP PRINTOCT

    POP BX
    RET
PRINT__OCT ENDP
                ;Εμφάνιση 8-bit δυαδικού αριθμού (από τον BL)
PRINT__BIN PROC NEAR
    MOV AL,BL
    MOV BL,2                 ;Δυαδικός => διαιρέσεις με 2
    MOV CX,0
GETBIN:
    MOV AH,0
    DIV BL
    PUSH AX
    INC CL
    CMP AL,0
    JNE GETBIN
PRINTBIN:
    POP AX
    ADD AH,48
    PRINTCH AH
    LOOP PRINTBIN

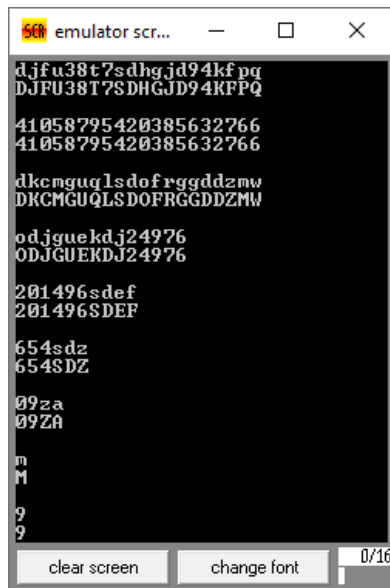
    RET
PRINT__BIN ENDP
CODE ENDS
END MAIN

```

4η Άσκηση

Το πρόγραμμα δέχεται έως 20 πεζούς αγγλικούς (a-z) και αριθμητικούς (0-9) χαρακτήρες από το πληκτρολόγιο και τους εμφανίζει στην οθόνη σε μία γραμμή τον έναν δίπλα στον άλλο. Με τη συμπλήρωση 20 χαρακτήρων ή όταν δοθεί *ENTER*, εμφανίζει στην επόμενη γραμμή την ίδια σειρά χαρακτήρων με τους αγγλικούς χαρακτήρες κεφαλαίους (A-Z). Το πρόγραμμα είναι συνεχούς λειτουργίας και τερματίζεται αν δοθεί =.

Κάθε χαρακτήρας εισάγεται, ελέγχεται για το αν ανήκει στα διαστήματα [0-9] και [a-z] και αν είναι δεκτός εμφανίζεται στην οθόνη και αποθηκεύεται στον πίνακα CHARS. Μόλις γεμίσει ο πίνακας CHARS ή όταν δοθεί *ENTER*, εμφανίζονται στην επόμενη γραμμή όλοι οι χαρακτήρες του με την ίδια σειρά, με τη διαφορά ότι οι πεζοί είναι πλέον κεφαλαίοι.



```
INCLUDE macros.asm
```

```
DATA SEGMENT
```

```
    CHARS DB 20 DUP(?)    ;Ο πίνακας των χαρακτήρων
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
    ASSUME CS:CODE, DS:DATA
```

```
MAIN PROC FAR
```

```
    MOV AX, DATA
```

```
    MOV DS, AX
```

```
    MOV CL, 0    ;Μετρητής ψηφίων
```

```
START:
```

```
    MOV DI, 0    ;Δείκτης πίνακα
```

```
NEXTCHAR: ;Εισαγωγή και έλεγχος χαρακτήρα
```

```
    READCH
```

```
    CMP AL, 61    ;= ?
```

```
    JE FINISH
```

```
    CMP AL, 13    ;ENTER ?
```

```
    JE CAPSLINE
```

```
    CMP AL, 48    ;<0 ?
```

```
    JB NEXTCHAR
```

```
    CMP AL, 122   ;>z ?
```

```
    JA NEXTCHAR
```

```
    CMP AL, 57    ;<=9 ?
```

```
    JBE SAVECHAR
```

```
    CMP AL, 97    ;<a ?
```

```
    JB NEXTCHAR
```

```
SAVECHAR:
```

```
    PRINTCH AL
```

```
    MOV CHARS[DI], AL ;Αποθήκευση
```

```
    INC DI
```

```
    INC CL
```

```
    CMP CL, 20    ;20 χαρακτήρες ?
```

```
    JB NEXTCHAR
```

```
CAPSLINE:
```

```
    PRINTLN
```

```
    CMP CL, 0    ;Έλεγχος κενού πίνακα
```

```
    JE NEXTCHAR ;
```

```
    MOV DI, 0    ;Μετατροπή και εμφάνιση
```

```
NEXTCAP:
```

```
    MOV AL, CHARS[DI]
```

```
    CMP AL, 97    ;<a ?
```

```
    JB SHOWCAPS
```

```
    CMP AL, 122   ;>z ?
```

```
    JA SHOWCAPS
```

```
    SUB AL, 32    ;Μικρό -> κεφαλαίο
```

```
SHOWCAPS:
```

```
    PRINTCH AL
```

```
    INC DI
```

```
    LOOP NEXTCAP
```

```
PRINTLN
```

```
PRINTLN
```

```
JMP START
```

```
FINISH:
```

```
    EXIT
```

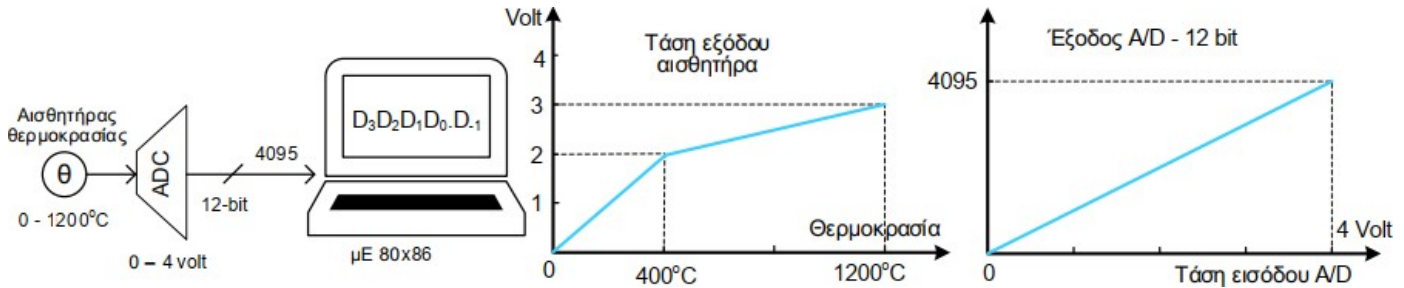
```
MAIN ENDP
```

```
CODE ENDS
```

```
END MAIN
```


5η Άσκηση

Το πρόγραμμα προσομοιώνει ένα σύστημα λήψης θερμοκρασίας που περιλαμβάνει έναν αισθητήρα θερμοκρασίας, έναν μετατροπέα από αναλογική τιμή σε ψηφιακή (ADC) και έναν υπολογιστή με τον $\mu\text{E } 80\text{x}86$. Υποτίθεται ότι ο αισθητήρας μετρά τη θερμοκρασία και παρέχει μία τάση στο διάστημα $[0,3]$ Volts στον ADC. Ο ADC ψηφιοποιεί την τάση του αισθητήρα στο διάστημα $[0,4095]$ Volts. Η ψηφιοποιημένη τάση παρέχεται ως είσοδος στον υπολογιστή, ο οποίος λαμβάνει τη θερμοκρασία μέσω μιας 16-bit θύρας εισόδου σε δυαδική μορφή των 12 bits και την απεικονίζει ως έξοδο στην οθόνη με έναν 4ψήφιο δεκαδικό αριθμό με ένα κλασματικό ψηφίο (XXXX,X) από 0,0 έως 1200,0 °C. Το σύστημα περιγράφεται από το παρακάτω σχήμα.



Η θύρα εισόδου προσομοιώνεται από το πληκτρολόγιο, μέσω του οποίου εισάγονται τα δεδομένα (η τάση του ADC) ως 3 δεκαεξαδικά ψηφία. Με την εκκίνηση της εκτέλεσης του προγράμματος εμφανίζεται το μήνυμα *START(Y,N)*: και ο χρήστης επιλέγει αν αυτό θα λειτουργήσει (Y) ή θα τερματιστεί (N). Σε περίπτωση λειτουργίας, το πρόγραμμα δέχεται τα 3 ψηφία της εισόδου (μόνο έγκυρα) και εμφανίζει τη θερμοκρασία. Το πρόγραμμα είναι συνεχούς λειτουργίας, τερματίζεται οποιαδήποτε στιγμή αν δοθεί ο χαρακτήρας N και σε περίπτωση θερμοκρασίας μεγαλύτερης από 1200,0 °C εμφανίζει το μήνυμα σφάλματος *ERROR*.

Το πρόγραμμα αρχικά εμφανίζει το μήνυμα εκκίνησης (*STARTPROMPT*) και τον χαρακτήρα που δίνει ο χρήστης. Κατά τη λειτουργία του, δέχεται τα 3 ψηφία της εισόδου στον AL με κλήση της ρουτίνας *HEX_KEYB* και τα ενώνει στον DX ολισθαίνοντάς τα κατάλληλα. Στη συνέχεια συγκρίνει την είσοδο με τα ψηφιοποιημένα άνω όρια των κλάδων της χαρακτηριστικής καμπύλης του αισθητήρα για να αποφασίσει σε ποιον κλάδο ανήκει και υπολογίζει τη θερμοκρασία υλοποιώντας την αντίστοιχη συνάρτηση. Για την υλοποίηση προγραμματιστικά των συναρτήσεων χρησιμοποιήθηκε η εντολή *DIV* που δίνει πηλίκο, άρα τα αποτελέσματα των υλοποιήσεων αυτών είναι τα ακέραια μέρη των ζητούμενων αριθμών και αποθηκεύονται στον AX. Από το υπόλοιπο της διαίρεσης, που αρχικά τοποθετείται στον DX, προκύπτει το μονοψήφιο κλασματικό μέρος των αριθμών. Οι συναρτήσεις των 2 κλάδων και ο τρόπος υπολογισμού των κλασματικών μερών φαίνονται παρακάτω.

$$\text{1ος κλάδος: } T = \frac{800V}{4095}$$

$$\text{2ος κλάδος: } T = \frac{3200V}{4095} - 1200$$

$$\text{κλασματικός μέρος} = \frac{10 * \text{υπόλοιπο}}{4095}$$

όπου T η ζητούμενη θερμοκρασία και V η τάση εξόδου του ADC. Επισημαίνεται ότι το κλασματικό μέρος είναι ίδιο και για τους 2 κλάδους, αφού έχουν τον ίδιο διαιρέτη στη συνάρτησή τους. Επισημαίνεται ακόμη ότι οι παραπάνω διαιρέσεις αναφέρονται σε ακέραια διαίρεση και ότι τα ψηφιοποιημένα άνω όρια των κλάδων της χαρακτηριστικής καμπύλης του αισθητήρα είναι οι τιμές της τάσης εξόδου του ADC για τις οποίες παίρνουμε θερμοκρασία όχι μεγαλύτερη από τις αντίστοιχες τιμές του οριζόντιου άξονα που φαίνονται στο σχήμα (άρα για τον 1ο κλάδο το 2 μετατρέπεται σε 2047 και για τον 2ο το 3 σε 3071). Η εμφάνιση του ακέραιου μέρους γίνεται μέσω του AX με κλήση της ρουτίνας *PRINT_DEC16* που τυπώνει έναν 16-bit δεκαδικό αριθμό και ακολουθείται από την εμφάνιση του κλασματικού μέρους.

```

501 emulator scr...
START(Y,N):Y
7FF 399.9
BFF 1199.8
BBD 1148.2
3E8 195.3
7D0 390.7
7D1 390.9
BB8 1144.3
9E7 780.9
8C6 555.1
B00 1000.5
400 200.0
7DE 393.4
AFB 996.6
C00 ERROR
C01 ERROR
FFF ERROR
BFF 1199.8
clear screen change font 0/16

```

```
INCLUDE macros.asm
```

```
DATA SEGMENT
```

```
STARTPROMPT DB "START(Y,N):$";Αρχικό μήνυμα  
ERRORMSG DB "ERROR$";Μήνυμα σφάλματος
```

```
ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS:CODE, DS:DATA
```

```
MAIN PROC FAR
```

```
MOV AX, DATA
```

```
MOV DS, AX
```

```
PRINTSTR STARTPROMPT
```

```
START: ;Εισαγωγή χαρακτήρα εκκίνησης
```

```
READCH
```

```
CMP AL, 'N' ;= N ?
```

```
JE FINISH ;Τερματισμός
```

```
CMP AL, 'Y' ;= Y ?
```

```
JE CONT ;Λειτουργία
```

```
JMP START
```

```
CONT:
```

```
PRINTCH AL ;Εμφάνιση χαρακτήρα εκκίνησης
```

```
PRINTLN
```

```
PRINTLN
```

```
NEWTEMP:
```

```
MOV DX, 0
```

```
MOV CX, 3 ;3 δεκαεξαδικά ψηφία
```

```
READTEMP: ;Είσοδος
```

```
CALL HEX_KEYB ;Εισαγωγή ψηφίου
```

```
CMP AL, 'N' ;Έλεγχος τερματισμού
```

```
JE FINISH
```

```
;Ένωση ψηφίων στον DX
```

```
PUSH CX
```

```
DEC CL ;Για την ολίσθηση
```

```
ROL CL, 2 ;
```

```
MOV AH, 0
```

```
ROL AX, CL ;Ολίσθηση αριστερά 8, 4, 0 ψηφία
```

```
OR DX, AX ;Προσθήκη ψηφίου στον αριθμό
```

```
POP CX
```

```
LOOP READTEMP
```

```
PRINTTAB
```

```
MOV AX, DX
```

```
CMP AX, 2047 ;V<=2 ?
```

```
JBE BRANCH1
```

```
CMP AX, 3071 ;V<=3 ?
```

```
JBE BRANCH2
```

```
PRINTSTR ERRORMSG ;V>3
```

```
PRINTLN
```

```
JMP NEWTEMP
```

```
BRANCH1: ;1ος κλάδος: V<=2, T=(800*V) div 4095
```

```
MOV BX, 800
```

```
MUL BX
```

```
MOV BX, 4095
```

```
DIV BX
```

```
JMP SHOWTEMP
```

```
BRANCH2: ;2ος κλάδος: 2<V<=3, T=((3200*V) div 4095)-1200
```

```
MOV BX, 3200
```

```
MUL BX
```

```
MOV BX, 4095
```

```
DIV BX
```

```
SUB AX, 1200
```

```
SHOWTEMP:
```

```
CALL PRINT_DEC16 ;Εμφάνιση ακέραιου μέρους (AX)
```

```
;Κλασματικό μέρος = (υπόλοιπο*10) div 4095
```

```
MOV AX, DX
```

```
MOV BX, 10
```

```
MUL BX
```

```
MOV BX, 4095
```

```
DIV BX
```

```
PRINTCH ', ' ;Υποδιαστολή
```

```

    ADD AL, 48          ;Κωδικός ASCII
    PRINTCH AL         ;Εμφάνιση κλασματικού μέρους
    PRINTLN
    JMP NEWTEMP

FINISH:
    PRINTCH AL
    EXIT
MAIN ENDP

HEX_KEYB PROC NEAR    ;Εισαγωγή δεκαεξαδικού ψηφίου (στον AL)
    READ              ;βλ. mP11_80x86_programs.pdf σελ. 20-21
    READCH
    CMP AL, 'N'       ;=N ?
    JE RETURN
    CMP AL, 48         ;<0 ?
    JL READ
    CMP AL, 57         ;>9 ?
    JG LETTER
    PRINTCH AL
    SUB AL, 48         ;Κωδικός ASCII
    JMP RETURN
LETTER:               ;A...F
    CMP AL, 'A'       ;<A ?
    JL READ
    CMP AL, 'F'       ;>F ?
    JG READ
    PRINTCH AL
    SUB AL, 55         ;Κωδικός ASCII
    RETURN:

    RET
HEX_KEYB ENDP

PRINT_DEC16 PROC NEAR ;Εμφάνιση 16-bit δεκαδικού αριθμού (από τον AX)
    PUSH DX            ;βλ. mP11_80x86_programs.pdf σελ. 26-27

    MOV BX, 10         ;Δεκαδικός => διαιρέσεις με 10
    MOV CX, 0          ;Μετρητής ψηφίων
GETDEC:               ;Εξαγωγή ψηφίων
    MOV DX, 0          ;Αριθμός mod 10 (υπόλοιπο)
    DIV BX             ;Διαίρεση με 10
    PUSH DX            ;Προσωρινή αποθήκευση
    INC CL
    CMP AX, 0          ;Αριθμός div 10 = 0 ? (πηλίκο)
    JNE GETDEC
PRINTDEC:             ;Εμφάνιση ψηφίων
    POP DX
    ADD DL, 48         ;Κωδικός ASCII
    PRINTCH DL
    LOOP PRINTDEC

    POP DX
    RET
PRINT_DEC16 ENDP
CODE ENDS
END MAIN

```

Σημείωση: Τα προγράμματα εκτελέστηκαν στον προσομοιωτή [emu8086 4.08](#), η επεξεργασία του σχήματος της 5ης Άσκησης έγινε στο [Paint](#) των Windows 10, η επεξεργασία του κώδικα των προγραμμάτων έγινε στο [Notepad++ 7.7 portable](#) και η συγγραφή της παρούσας αναφοράς έγινε στο [LibreOffice 6.3.1 portable](#).