ΣΗΜΜΥ ΕΜΠ / ΡΟΗ Υ
Συστήματα Μικροϋπολογιστών / 5η ομάδα ασκήσεων
Νικόλαος Πηγαδάς / el18445
2020-2021

Άσκηση 1

```
;=========================================
;                    MACROS
;=========================================

;Typwnei character
PRINTCH MACRO CHAR
   PUSH AX
   PUSH DX
   MOV DL,CHAR
   MOV AH,2
   INT 21H
   POP DX
   POP AX
ENDM

;Allagh grammhs
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

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

```
;=============================================
;                    SEGMENTS
;=============================================
DATA SEGMENT
   TABLE DB 128 DUP(?)   ;Pinakas dedomenwn
   TWO DB DUP(2)         ;Gia ton elegxo ths isotimias twn arithmwn
DATA ENDS


CODE SEGMENT
ASSUME CS:CODE, DS:DATA


;=============================================
;                     MAIN
;=============================================
MAIN PROC FAR
   MOV AX,DATA
   MOV DS,AX

;.......Apothikeush arithmwn sthn mnhmh.......
   MOV DI,0              ;Deikths ston pinaka arithmwn
   MOV CX,128            ;Plhthos arithmwn

STORE:
   MOV TABLE[DI],CL
   INC DI
   LOOP STORE

;....Elegxos isotimias, arthroish, metrhsh....
   MOV DH,0              ;Gia thn prosthesh AX+DL
   MOV AX,0              ;Athroisma perittwn
   MOV BX,0              ;Plhthos perittwn
   MOV DI,0
   MOV CX,128

ADDODD:
   PUSH AX
   MOV AH,0              ;Gia thn diairesh AX/2
   MOV AL,TABLE[DI]      ;Elegxos isotimias
   DIV TWO
   CMP AH,0
   POP AX
   JE SKIPEVEN           ;An o arithmos ston AX einai artios kane skip
   MOV DL,TABLE[DI]      ;Proswrinh apoth_hkeush
```

```
   ADD AX,DX                ;Prosthesh
   INC BX                   ;Aukshsh metrhth perittwn

SKIPEVEN:                   ;Skip an einai artios
   INC DI
   LOOP ADDODD

;.........Ypologismos kai ektypwsh MO.........
   MOV DX,0
   DIV BX                   ;Diairesh me to plhthos tous

   CALL PRINT_NUM8_HEX ;Ektypwsh akeraiou merous phlikou
   PRINTLN

;..............Megisto & Elaxisto.............
   MOV AL,TABLE[0]          ;Arxikopoihsh max
   MOV BL,TABLE[127]        ;Arxikopoihsh min
   MOV DI,0
   MOV CX,128

MAXMIN:
   CMP AL,TABLE[DI]         ;Elegxos parontos stoixeiou
   JC NEWMAX
   JMP MIN

NEWMAX:                     ;An einai megalutero tou max -> new max
   MOV AL,TABLE[DI]
   JMP NEXTNUM

MIN:                        ;Elegxos min
   CMP TABLE[DI],BL
   JC NEWMIN
   JMP NEXTNUM

NEWMIN:                     ;An einai mikrotero tou min -> new min
   MOV BL,TABLE[DI]

NEXTNUM:                    ;Epomenos arithmos
   INC DI
   LOOP MAXMIN
   CALL PRINT_NUM8_HEX
   PRINTCH ' '
   MOV AL,BL
   CALL PRINT_NUM8_HEX
EXIT

MAIN ENDP
```

```
;Ektypwsh 8-bit hex arithmou
PRINT_NUM8_HEX PROC NEAR;
   MOV DL,AL
   AND DL,0F0H              ;1o hex
   MOV CL,4
   ROR DL,CL
   CMP DL,0                 ;Agnohsh arxikou 0
   JE SKIPZERO
   CALL PRINT_HEX
SKIPZERO:
   MOV DL,AL
   AND DL,0FH               ;2o hex - apomonwsh tou ston DL
   CALL PRINT_HEX
   RET
PRINT_NUM8_HEX ENDP


;Ektypwsh hex pshfiou apo diafaneies
PRINT_HEX PROC NEAR    ;Ektypwsh periexomenwn DL
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
```

Για την δοκιμή του παραπάνω προγράμματος έγινε χρήση του emulator του 8086:

emu8086

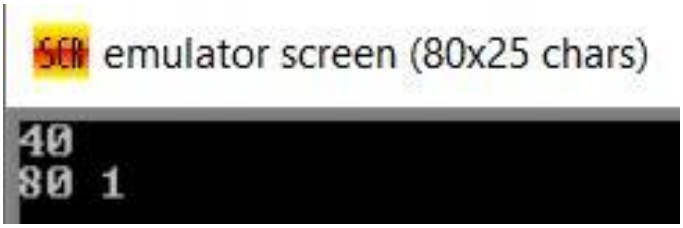
https://emu8086-microprocessor-emulator.en.softonic.com/

Παρακάτω παρατίθενται τα αποτελέσματα του προγράμματος, που είναι και τα επιθυμητά:

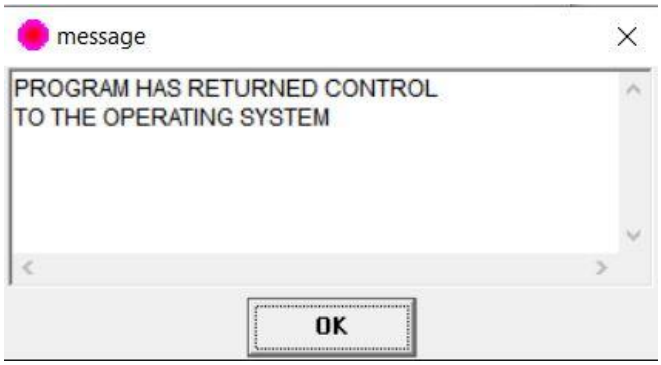
MO = 40_{hex} = 64_{dec}

$$(= \frac{4096_{dec}}{64_{dec}} = \frac{\text{άθροισμα περιττών ως το 128}}{\text{πλήθος περιττών ως το 128}})$$

max = 80_{hex} = 128_{dec} **min** = 1_{hex} = 1_{dec}



Κατόπιν, εμφανίζεται μήνυμα ότι ο έλεγχος επιστρέφει στο λειτουργικό σύστημα:



Άσκηση 2

```
;=============================================
;                    MACROS
;=============================================


;Typwnei character
PRINTCHAR MACRO CHAR
  PUSH AX
  PUSH DX
  MOV DL,CHAR
  MOV AH,2
  INT 21H
  POP DX
  POP AX
ENDM

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

;Allagh grammhs
PRINTLN MACRO
  PUSH AX
  PUSH DX
  MOV DL,0AH
  MOV AH,2
  INT 21H
  MOV DL,0DH
  MOV AH,2
  INT 21H
  POP DX
  POP AX
ENDM
```

```
;Diavasma character
READCHAR MACRO
  MOV AH,8
  INT 21H
ENDM

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


;=============================================
;                          SEGMENTS
;=============================================
DATA SEGMENT
   PRINTZ DB 'Z=$'          ;Typwnei 'Z=' kai perimenei eisodo
   PRINTW DB 'W=$'          ;Typwnei 'W=' kai perimenei eisodo
   PRINTSUM DB 'Z+W=$'  ;Typwnei 'Z+W=' kai typwnei to apotelesma
                            ;me vash tis parapanw eisodous
   PRINTDIFF DB 'Z-W=$'  ;Typwnei 'Z-W=' kai typwnei to apotelesma
                            ;me vash tis parapanw eisodous
   Z DB 0
   W DB 0
DATA ENDS

CODE SEGMENT
        ASSUME CS:CODE, DS:DATA


;=============================================
;                          MAIN
;=============================================
MAIN PROC FAR
                MOV AX,DATA
                MOV DS,AX

        START:
                PRINTSTR PRINTZ
                CALL READDIG       ;Diavase to prwto pshfio tou Z
     MOV BL, 10
           MUL BL
                LEA DI,Z              ;Swse to
                MOV [DI],AL
           CALL READDIG             ;Diavase to deutero pshfio tou Z
```

```
                ADD [DI],AL          ;Swse to

                PRINTCHAR ' '        ;Typwnei keno metaksh twn 'Z=$' kai 'W=$'

                PRINTSTR PRINTW
            CALL READDIG                    ;Diavase to prwto pshfio touf W
    MOV BL, 10
            MUL BL
            LEA DI,W                        ;Swse to
                MOV [DI],AL
                CALL READDIG                ;Diavase to deutero pshfio tou W
                ADD [DI],AL                 ;Swse to

                PRINTLN

            MOV AL,[DI]                      ;     W
                LEA DI,Z                     ;     Z
                ADD AL,[DI]                  ;Ypologismos athroismatos

                PRINTSTR PRINTSUM
                CALL PRINT_8BIT_HEX          ;Typwse to apotelesma

                PRINTCHAR ' '

                MOV AL,[DI]                  ;     Z
                LEA DI,W                     ;     W
                MOV BL,[DI]

    PRINTSTR PRINTDIFF
                CMP AL,BL                    ;Sygkrish Z me W...
                JB NEG                       ;...gia na doume an to apotelesma
                                             ;einai arnhtiko h thetiko
                SUB AL,BL                    ;An Z>W
                JMP PRINTSUB

        NEG:
                SUB BL,AL                    ;An Z<W
                MOV AL,BL
    PRINTCHAR '-'

        PRINTSUB:
                        CALL PRINT_8BIT_HEX  ;Typwse to apotelesma
                        PRINTLN
                        PRINTLN
                        JMP START
MAIN ENDP
```

```
;..Diavase-typwse-apoth_hkeuse ena decadiko pshfio ston AL..
READDIG PROC NEAR
        READ:
                        READCHAR
                        CMP AL,48
                        JB READ
                        CMP AL,57          ;Diavaze thn eisodo mexri na
                                           ;einai metaksu 0 kai 9
                        JA READ
                        PRINTCHAR AL
                        SUB AL,48          ;Kwdikas ASCII
                        RET
READDIG ENDP


;.........Print 8-bit number in HEX (saved in AL).........
PRINT_8BIT_HEX PROC NEAR
                        MOV DL,AL
                        AND DL,0F0H        ;Apomonwse to prwto hex pshfio
                        MOV CL,4
                        ROR DL,CL
                        CMP DL,0           ;An einai 0, mhn to typwseis
                        JE ZERO
                        CALL PRINT_HEX
        ZERO:
                        MOV DL,AL
                        AND DL,0FH         ;Apomonwse to deutero hex pshfio
                        CALL PRINT_HEX
                        RET
PRINT_8BIT_HEX ENDP


;................Print HEX digit (saved in DL).................
PRINT_HEX PROC NEAR
                        CMP DL,9       ;An o arithmos einai metaksy twn 0 kai 9,
                                       ;prosthese 30 hex
                        JG ADDR1
                        ADD DL,30H
                        JMP ADDR2
        ADDR1:
                        ADD DL,37H  ;Alliws an o arithmos einai metaksu twn A kai F,
                                    ;prosthese 37 hex
        ADDR2:
                        PRINTCHAR DL
                        RET
PRINT_HEX ENDP
```

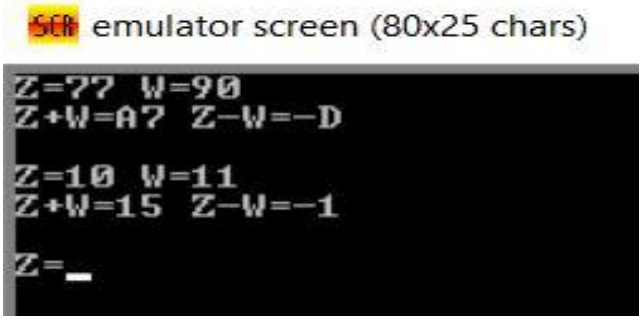
```
CODE ENDS
END MAIN
```

Για την δοκιμή του παραπάνω προγράμματος έγινε χρήση του emulator του 8086:

emu8086

https://emu8086-microprocessor-emulator.en.softonic.com/

Παρακάτω παρατίθενται δύο παραδείγματα δοκιμών εκτέλεσης, μεταξύ των διαφόρων που δοκιμάστηκαν για την ορθότητα του προγράμματος:



Συγκεκριμένα, για εισόδους τους δεκαδικούς αριθμούς $Z = 77_{dec}$ και $W = 90_{dec}$ έχουμε το ορθό αποτέλεσμα της πρόσθεσης, καθώς και της αφαίρεσής τους στο δεκαεξαδικό σύστημα:

$$Z + W = A7_{hex} = 167_{dec}$$
$$Z - W = -D_{hex} = -13_{dec}$$

Ομοίως για $Z = 10_{dec}$ και $W = 11_{dec}$:

$$Z + W = 15_{hex} = 21_{dec}$$
$$Z - W = -1_{hex} = -1_{dec}$$

και άρα επαληθεύεται η σωστή λειτουργία του ζητούμενου προγράμματος. Τέλος, αξίζει να επισημανθεί ότι το πρόγραμμα είναι συνεχούς λειτουργίας και άρα δεν τερματίζει, αλλά περιμένει επόμενη είσοδο.

Άσκηση 3

```
;=========================================
;                    MACROS
;=========================================


READ MACRO
   MOV AH, 08H
   INT 21H
ENDM

PRINT MACRO CHAR
   MOV DL, CHAR
   MOV AH, 02H
   INT 21H
ENDM

PRINT_STR MACRO STRING
   MOV DX, OFFSET STRING
   MOV AH, 09H
   INT 21H
ENDM

EXIT MACRO
   MOV AX, 4C00H
   INT 21H
ENDM


;=========================================
;                    MAIN
;=========================================
START:
        MOV CX, 3           ; Arxikopoihsh metrhth se 3 gia ta 3 hex psifia
        MOV BX, 0000H       ; BX = 0000000000000000
                            ; Ston opoio tha ginei apoth_hkeush tou pshfiou eisodou

INPUT:
        CALL HEX_KEYB       ; Diavasma hex pshfiou h termatikou kai apoth_hkeush
ston AL
        CMP AL, 'T'         ; An einai to termatiko 'T'...
```

```
        JE QUIT             ; ...to programma teleiwnei
        OR BX, AX           ; Alliws apoth_hkeush tou hex artihmou ston BX
        SHL BX, 1           ; 4 aristeres olisth_hseis
        SHL BX, 1           ; oses osa ta binary bits enos hex psifiou
        SHL BX, 1
        SHL BX, 1
        LOOP INPUT          ; Diavasma eisodou mexri to periexomeno tou CX na
ginei 0

OUTPUT:
        MOV CL, 4           ; 4 dekdies olisth+_hseis tou BX...
        SHR BX, CL          ; ...giati olisth_hsame thn eisodo 4 fores aristera
        CALL PRINT_HEX      ; Typwnei ton isodynamo hex arithmo
  PRINT "="
  CALL PRINT_DEC        ; Typwnei ton isodynamo dec arithno
  PRINT "="
  CALL PRINT_OCT        ; Typwnei ton isodynamo oct arithmo
  PRINT "="
  CALL PRINT_BIN        ; Typwnei ton isodynamo bin arithmo
  MOV DX, 0AH           ; Nea grammh
  MOV AH, 02H
  INT 21H
  MOV DX, 0DH
  MOV AH, 02H
  INT 21H
        JMP START           ; Epanalaves

QUIT:
        EXIT                ; Telos programmatos

HEX_KEYB PROC NEAR      ; Yporoutina pou diavazei egkyro hex pshfio apo thn
eisodo...
                ; ...kai ton apoth_hkeuei AL
        IGNORE:
                READ            ; Diavasma eisodou kai apoth_hkeush ASCII timhs ston
AL
                MOV AH, 00H     ; AH = 00000000
                CMP AL, 'T'     ; Elegxos gia termatiko symvolo
                JE FINISH       ; An vrethei, h yporoytina termatizei
                CMP AL, 30H     ; An timh ASCII <30 hex mh egkyro hex pshfio...
                JL IGNORE       ; ...diavase ksana thn eisodo
                CMP AL, 39H     ; An timh ASCII metaksu 30 hex and 39 hex [kleisto
synolo]
                JBE FOO_1       ; tote to pshfio einai egkyro
                CMP AL, 40H     ; An timh ASCII 40 hex, den eimai egkyro...
                JE IGNORE       ; ...kai diavazoume ksana
                CMP AL, 46H     ; An timh ASCII megalyterh apo 46H
```

```
                JA IGNORE          ; tote to pshfio den einai egkyro...
                                   ; ...kai diavazoume ksana
                SUB AL, 31H        ; AN timh ASCII metaksy 41 hex kai 46 hex [kleisto
synolo]
                SUB AL, 06H        ; tote to pshfio einai ena apo ta: A, B, C, D, E, F
                                   ; Afairoume 31 hex apo thn ASCII timh...
                                   ; ...kai 6 hex gia dekadiko apotelesma
                JMP FINISH

   FOO_1:
                SUB AL, 30H        ; Pshfio metaksy 0 and 9 [kleisto synolo]
                                   ; Afairw 30 hex apo thn ASCII timh
         FINISH:
                RET
HEX_KEYB ENDP

; Typwnei 12-bit hex arithmo
PRINT_HEX PROC NEAR
   PUSH BX
   PUSH CX
   PUSH DX

   MOV DX, BX
   AND DX, 0F00H          ; Apomonwsh shmantikou hex pshfiou
   CMP DH, 09H            ; Evresh ASCII timhs
   JBE ADDR1
   ADD DH, 0031H
   ADD DH, 06H
   JMP ADDR2
   ADDR1:
      ADD DH, 0030H
   ADDR2:
      PRINT DH

   MOV CX, 4
   MOV DX, BX
   AND DX, 00F0H          ; Apomonvsh deuterou pio shmantikou pshfiou tou hex
   SHL DX, CL             ; Evresh ASCII timhs
   CMP DH, 09H
   JBE ADDR3
   ADD DH, 0031H
   ADD DH, 06H
   JMP ADDR4
   ADDR3:
      ADD DH, 0030H
   ADDR4:
      PRINT DH
```

```
   MOV DX, BX
   AND DX, 000FH          ; Apomonwsh tritou kai ligoterou shmantikou hex pshfiou
   CMP DL, 09H            ; Evresh ASCII timhs
   JBE ADDR5
   ADD DL, 0031H
   ADD DL, 06H
   JMP ADDR6
   ADDR5:
      ADD DL, 0030H
   ADDR6:
      PRINT DL

   POP DX
   POP CX
   POP BX
   RET
PRINT_HEX ENDP


; Typwnei 12-bit dec arithmo
PRINT_DEC PROC NEAR
   PUSH BX
   PUSH CX

   MOV DX, BX

   MOV CX, 00FFH

; Xiliades
   FOR1:
      INC CX
      SUB DX, 03E8H
      CMP DX, 0000H
      JGE FOR1
   ADD DX, 03E8H

   PUSH DX
   ADD CL, 30H            ; ASCII
   PRINT CL               ; Typwnei xiliades
   POP DX

   MOV CX, 00FFH

; Ekatontades
   FOR2:
      INC CX
```

```
      SUB DX, 0064H
      CMP DX, 0000H
      JGE FOR2
   ADD DX, 0064H

   PUSH DX              ; ASCII
   ADD CL, 30H          ; Typwnei ekatontades
   PRINT CL
   POP DX

   MOV CX, 00FFH

; Decades
   FOR3:
      INC CX
      SUB DX, 000AH
      CMP DX, 0000H
      JGE FOR3
   ADD DX, 000AH

   PUSH DX
   ADD CL, 30H          ; ASCII
   PRINT CL             ; Typwnei decades
   POP DX

; Monades
   PUSH DX
   ADD DL, 30H          ; ASCII
   PRINT DL             ; Typwnei monades
   POP DX

   POP CX
   POP BX
   RET
PRINT_DEC ENDP

; Typwnei 12-bit oct arithmo
PRINT_OCT PROC NEAR
   PUSH BX
   PUSH AX
   PUSH DX

   MOV DX, BX           ; Apomonwsh shmantikoterou oct pshfiou
   AND DH, 0EH          ; Evresh ASCII timhs
   SHR DH, 1
   ADD DH, 30H
   PRINT DH
```

```
   MOV DX, BX
   MOV CL, 2
   AND DX, 01C0H          ; Apomonwsh deuterou shmantikoterou oct pshfiou
   SHL DX, CL             ; Evresh ASCII timhs
   ADD DH, 30H
   PRINT DH

   MOV DX, BX
   MOV CL, 3
   AND DL, 38H            ; Apomonwsh tritou shmantikoterou oct pshfio
   SHR DL, CL             ; Evresh ASCII timhs
   ADD DL, 30H
   PRINT DL

   MOV DX, BX
   AND DL, 07H            ;  Apomonwsh tetartou kai ligoterou shmantikou oct pshfio
   ADD DL, 30H            ; Evresh ASCII timhs
   PRINT DL

   POP DX
   POP AX
   POP BX
   RET
PRINT_OCT ENDP

; Typwnei 12-bit bin arithmo
PRINT_BIN PROC NEAR

   PUSH BX
   PUSH AX

   MOV CL, 4
   SHL BX, CL

   MOV CX, 12             ; CX metrhths gia ta 12 bits

   FOO:                   ; Elegxos twn 12 bits
                          ; An bit = 0, tote ASCII: 30 hex // An bit = 1, tote ASCII: 31 hex
      RCL BX, 1           ; Mia aristerh olisth_hsh, symperilamvanomenou kratoumenou
      JC L
      MOV AL, 30H
      JMP CONT
      L:
         MOV AL, 31H
      CONT:
         PRINT AL
```

```
        LOOP FOO        ; Epanalave mexri na mhdenistei o metrhths

   POP AX
   POP BX
   RET
PRINT_BIN ENDP

RET
```

Παρακάτω παρατίθενται διάφορα αποτελέσματα εκτελέσεων του προγράμματος, συμπεριλαμβανομένης της περίπτωσης λήψης του τερματικού χαρακτήρα ‘T’, κατά την οποία τερματίζεται το πρόγραμμα και επιστρέφεται ο έλεγχος στο λειτουργικό σύστημα:

Άσκηση 4

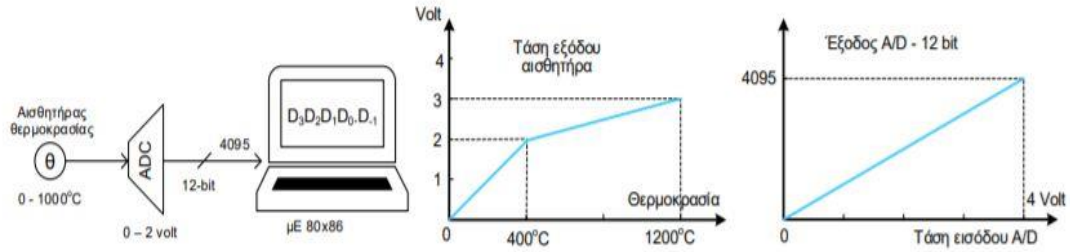
```
;=============================================
;                    MACROS
;=============================================
PRINT MACRO CHAR
   PUSH AX
   PUSH DX
   MOV DL,CHAR
   MOV AH,2
   INT 21H
   POP DX
   POP AX
ENDM PRINT


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


PRNT_STR MACRO STRING
   MOV DX,OFFSET STRING
   MOV AH,9
   INT 21H
ENDM PRNT_STR


READ MACRO
   MOV AH,8
   INT 21H
ENDM READ


;=============================================
;                   SEGMENTS
;=============================================
DATA_SEG    SEGMENT
   SYMBOLS DB 20 DUP(?)
   NEWLINE DB 0AH,0DH,'$'
DATA_SEG    ENDS
```

```
CODE_SEG    SEGMENT
   ASSUME CS:CODE_SEG, DS:DATA_SEG


;=============================================
;                              MAIN
;=============================================
MAIN PROC FAR

   MOV AX,DATA_SEG
   MOV DS,AX

START:
   MOV CX,20          ;Arxikopoihsh metrhth gia 20 chars
   MOV DI,0           ;Arxikopoihsh index pinaka
INPUT:
   READ               ;Anagnwsh eisodou
   CMP AL,61                     ;An eisodos '=', tote exoume termatismo
   JE END_OF_PROG
   CMP AL,0DH         ;An h eisodo einai enter exoume diakoph eisodou
   JE PRINT_INPUT
   CMP AL,30H         ;An eisodos <30 hex, diavasma neas eisodou
   JL INPUT
   CMP AL,39H         ;An eisodos >39 hex, tote isws gramma...
   JG MAYBE_LETTER
   JMP ACCEPTABLE_INPUT ;...alliws arithmos

MAYBE_LETTER:
   CMP AL,'a'         ;An eisodos <'a', diavasma neas eisodou
   JL INPUT
   CMP AL,'z'         ;An eisodos >'z', diavasma neas eisodou
   JG INPUT

ACCEPTABLE_INPUT:       ;Apoth_hkeush eisodou
   MOV [SYMBOLS+DI],AL
   INC DI
   LOOP INPUT         ;An CX=/=0, diavasma neas eisodou

PRINT_INPUT:
   MOV CX,20
   MOV DI,0
OUTPUT:
   PRINT [SYMBOLS+DI]   ;Typwma charakthrwn
   INC DI
   LOOP OUTPUT
```

```
   PRNT_STR NEWLINE


   MOV CX,20
   MOV DI,0

PRINT_LETTERS:          ;Typwma charakthrwn gia arxh
   MOV AL,[SYMBOLS+DI]
   CMP AL,'a'           ;Elegxos an to gramma einai pezo
   JL  NOT_A_LETTER
   CMP AL,'z'
   JG NOT_A_LETTER
   SUB AL,32            ;Kanto kefalaio
   PRINT AL             ;Typwma kefalaiwn

NOT_A_LETTER:           ;An den einai gramma, diavasma neas eisodou
   INC DI
   LOOP PRINT_LETTERS

   PRINT '-'

   MOV CX,20
   MOV DI,0

PRINT_NUMS:             ;Typwma arithmwn
   MOV AL,[SYMBOLS+DI]
   CMP AL,30H           ;Elegxos an exoume arithmo
   JL  NOT_A_NUMBER
   CMP AL,39H
   JG NOT_A_NUMBER
   PRINT AL             ;Typwma mono arithmwn

NOT_A_NUMBER:           ;An den einai arithmos, diavasma neas eisodou
   INC DI
   LOOP PRINT_NUMS

   PRNT_STR NEWLINE
   JMP START

END_OF_PROG:
   EXIT
MAIN ENDP


CODE_SEG ENDS
END MAIN
```

Παρακάτω παρατίθενται διάφορα αποτελέσματα εκτελέσεων του προγράμματος, συμπεριλαμβανομένης της περίπτωσης λήψης του τερματικού χαρακτήρα ‘=’, κατά την οποία τερματίζεται το πρόγραμμα και επιστρέφεται ο έλεγχος στο λειτουργικό σύστημα:

Άσκηση 5



Συμπληρωματικά σχόλια

Το παρακάτω πρόγραμμα επεξεργάζεται δεδομένα ενός συστήματος που αποτελείται από έναν αισθητήρα θερμοκρασίας, έναν μετατροπέα ADC (από αναλογική τιμή σε ψηφιακή) και ένα υπολογιστικό σύστημα με τον μΕ 8086. Ο αισθητήρας μετρά τη θερμοκρασία και παρέχει μία τάση στο διάστημα 0 έως 2 Volts στον ADC, ο οποίος την ψηφοποιεί στο διάστημα [0,4095] Volts και την παρέχει σαν είσοδο στον υπολογιστή. Ο υπολογιστής, λοιπόν, 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. Οι δεκαδικές τιμές που απεικονίζονται μπορούν να κυμαίνονται από 0,0 έως 1200,0 °C, αλλιώς έχουμε ERROR.

Η θύρα εισόδου προσομοιώνεται από το πληκτρολόγιο, μέσω του οποίου εισάγονται τα δεδομένα, η τάση του ADC, ως 3 hex ψηφία. Αρχικά, το πρόγραμμα εμφανίζει το μήνυμα ‘START(Y,N):’ και ο χρήστης καλείται να αποφασίσει αν αυτό θα λειτουργήσει (‘Y’) ή θα τερματιστεί (‘N’). Σε περίπτωση λειτουργίας, το πρόγραμμα δέχεται τα 3 έγκυρα ψηφία της εισόδου (μέσα στα επιτρεπόμενα πλαίσια) και εμφανίζει τη θερμοκρασία στην οθόνη. Το πρόγραμμα είναι συνεχούς λειτουργίας. Τερματίζεται μόνο αν δοθεί ο χαρακτήρας ‘N’.

Ειδικότερα, σε επίπεδο προγραμματισμού του 8086, το πρόγραμμα αρχικά εμφανίζει μήνυμα εκκίνησης (STARTPROMPT). Δέχεται τον χαρακτήρα που δίνει ο χρήστης ‘Y’ ή ‘N’. Κατά τη λειτουργία του, λαμβάνει τα 3 ψηφία της εισόδου στον AL με κλήση της ρουτίνας HEX_KEYB. Κατόπιν, τα τοποθετεί στον DX, κάνοντας τις απαραίτητες ολισθήσεις.

Ύστερα, γίνεται η σύγκριση της εισόδου με τα ψηφιοποιημένα άνω όρια των κλάδων της χαρακτηριστικής καμπύλης του αισθητήρα για την επιλογή του αντίστοιχου κλάδου, στον οποίο ανήκει. Η θερμοκρασία υπολογίζεται υλοποιώντας την αντίστοιχη συνάρτηση.

Η υλοποίηση των συναρτήσεων επιτεύχθηκε μέσω της εντολή DIV που δίνει τα ακέραια μέρη των θερμοκρασιών και αποθηκεύονται στον AX. Από το υπόλοιπο της διαίρεσης, το οποίο τοποθετείται στον DX, προκύπτει το μονοψήφιο κλασματικό μέρος των αριθμών.

Οι συναρτήσεις των 2 κλάδων και ο τρόπος υπολογισμού των κλασματικών μερών φαίνονται παρακάτω:

1ος κλάδος: $$T = \frac{800}{4095}$$

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

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

όπου T η ζητούμενη θερμοκρασία και V η τάση εξόδου του ADC.

Εδώ αξίζει να αναφερθεί ότι το κλασματικό μέρος είναι ίδιο και για τους 2 κλάδους, αφού έχουν τον ίδιο διαιρέτη στη συνάρτησή τους.

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

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

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

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

```
READCH MACRO
   MOV AH,8
   INT 21H
ENDM

EXIT MACRO
   MOV AX,4C00H
   INT 21H
ENDM


;=============================================
;                    SEGMENTS
;=============================================
DATA SEGMENT
      STARTPROMPT DB "START(Y,N):$"
      ERRORMSG DB "ERROR$"
ENDS

CODE SEGMENT
   ASSUME CS:CODE, DS:DATA


;=============================================
;                      MAIN
;=============================================
MAIN PROC FAR
   MOV AX,DATA
   MOV DS,AX
   PRINTSTR STARTPROMPT   ;Tupwma "START(Y,N):" kai perimenoume eisodo

START:
   READCH              ;Diavasma eisodou
   CMP AL,'N'
   JE FINISH           ;An h eisodos einai N tote exoume termatismo
   CMP AL,'Y'
   JE CONT             ;An einai Y synexizoume
   JMP START

CONT:
   PRINTCH AL          ;Typwma Y h N, analogws ti edwse o xrhsths
   PRINTLN
   PRINTLN

NEWTEMP:
   MOV DX,0
   MOV CX,3            ;Metrhths triwn hex pshfiwn
```

```
READTEMP:
   CALL HEX_KEYB        ;Diavasma eisodou
   CMP AL,'N'           ;Elegxos gia N, alliws -pithanws- exoume hex pshfio
   JE FINISH

;.......Enwsh pshfiwn ston DX.......;
   PUSH CX
   DEC CL               ;Meiwsh metrhth//analogws thn shmasia theshs pshfiou...
                        ;...kanoume antistoixa toses olisth_hseis
   ROL CL,2             ;2 aristeres olisth_hseis
   MOV AH,0
   ROL AX,CL            ;Aristeres olisth_hseis, oses o metrhths -> 8 4 0 pshfia
   OR DX,AX             ;Vazoume to pshfio ston arithmo
   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:                ;1os klados V<=2, T=(800*V) div 4095
   MOV BX,800
   MUL BX
   MOV BX,4095
   DIV BX
   JMP SHOWTEMP

BRANCH2:                ;2os klados: 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     ;Akeraio meros (AX)
                        ;Klasmatiko meros = (upoloipo * 10) div 4095
   MOV AX,DX
   MOV BX,10
   MUL BX
   MOV BX,4095
```

```
   DIV BX
   PRINTCH ','          ;Ypodiastolh
   ADD AL,48            ;ASCII
   PRINTCH AL           ;Klasmatiko meros
   PRINTLN
   JMP NEWTEMP

FINISH:
PRINTCH AL
EXIT

MAIN ENDP


HEX_KEYB PROC NEAR
READ:
   READCH
   CMP AL,'N'           ;=N ?
   JE RETURN
   CMP AL,48            ;<0 ?
   JL READ
   CMP AL,57            ;>9 ?
   JG LETTER
   PRINTCH AL
   SUB AL,48            ;ASCII
   JMP RETURN

LETTER:                 ;A...F
   CMP AL,'A'           ;<A ?
   JL READ
   CMP AL,'F'           ;>F ?
   JG READ
   PRINTCH AL
   SUB AL,55
                    ;ASCII
RETURN:
   RET
HEX_KEYB ENDP


PRINT_DEC16 PROC NEAR
   PUSH DX
   MOV BX,10            ;Diaireseis me 10 gia decadiko
   MOV CX,0             ;Metrhths pshfiwn
GETDEC:                 ;Eksagwgh pshfiwn
   MOV DX,0             ;Ypoloipo (arithmos mod 10)
   DIV BX               ;Diairesh me 10
```

```
   PUSH DX            ;Apothh_hkeush arithmou (proswrinh)
   INC CL
   CMP AX,0           ;An piliko=/=0  ((arithmosdiv10) =/= 0)...
   JNE GETDEC          ;...pame sto GETDEC -> Afairese 10
PRINTDEC:             ;Emfanish pshfiwn
   POP DX
   ADD DL,48          ;ASCII
   PRINTCH DL
   LOOP PRINTDEC
   POP DX
   RET

PRINT_DEC16 ENDP
CODE ENDS

END MAIN
```

Παρακάτω παρατίθενται διάφορα αποτελέσματα εκτελέσεων του προγράμματος, συμπεριλαμβανομένης της περίπτωσης λήψης του τερματικού χαρακτήρα ‘N’, κατά την οποία τερματίζεται το πρόγραμμα και επιστρέφεται ο έλεγχος στο λειτουργικό σύστημα: