Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΑΚΑΔ. ΕΤΟΣ 2019-2020

Συστήματα Μικροϋπολογιστών, Ροή Υ, εξάμηνο 6ο

5η Σειρά Αναλυτικών Ασκήσεων

Στοιχεία φοιτητών:

Όνομα: Μαντζόυτας Ανδρέας, ΑΜ: 03117108, Εξάμηνο: 6ο

Όνομα: Τσιτσής Αντώνιος , ΑΜ: 03117045, Εξάμηνο: 6ο

Ημερομηνία Παράδοσης: 17/07/2020

1η Άσκηση

```
INCLUDE MACROS.ASM
   .8086
   .MODEL SMALL
   .STACK 256

DATA SEGMENT
DATA ENDS

CODE SEGMENT
   ASSUME CS:CODE, DS: DATA
MAIN PROC FAR
            MOV AX,DATA
            MOV DS,AX
START:
    CALL HEX_KEYB
    CMP AL,'Q'
    JE FINISH
    MOV BL,16
    MUL BL
    MOV BL,AL                   ; o bl exei to proto psifio

    CALL HEX_KEYB
    CMP AL,'Q'
    JE FINISH
    ADD BL,AL                               ;o arithmos einai ston bl

    CALL PRINT_HEX
    CALL PRINT_DECI
    CALL PRINT_OCT
    CALL PRINT_BIN

    JMP START
FINISH:
   EXIT
MAIN ENDP


HEX_KEYB PROC NEAR
   PUSH DX
IGNORE:
   READ
   CMP AL,'Q'
   JE ADDR2
   CMP AL,30H
   JL IGNORE
   CMP AL,39H
   JG ADDR1
   SUB AL,30H
   JMP ADDR2
ADDR1:
   CMP AL,'A'
   JL IGNORE
   CMP AL,'F'
   JG IGNORE
   SUB AL,37H
ADDR2:
   POP DX
   RET
```

```
HEX_KEYB ENDP

PRINT_DECI PROC NEAR
   PUSH BX
   MOV AH,0
   MOV AL,BL
   MOV BL,10
   MOV CX,0
DEC_ADDR:
   DIV BL
   INC CX
   PUSH AX
   CMP AL,0          ; an den uparxoun alles 10ades, exo mono monades
   JE DEC_PRINT
   MOV AH,0
   JMP DEC_ADDR
DEC_PRINT:
   POP DX
   MOV DL,DH
   MOV DH,0
   ADD DX,30H  ; to kanoume ascii
   MOV AH,2
   INT 21H
   LOOP DEC_PRINT          ; loop mexri na tupothoun ola
   POP BX
   PRINT '='
   RET
ENDP PRINT_DECI

PRINT_OCT PROC NEAR
   PUSH BX
   MOV AH,0
   MOV AL,BL
   MOV BL,8
   MOV CX,0
OCT_ADDR1:
   DIV BL
   INC CX
   PUSH AX
   CMP AL,0
   JE OCT_ADDR2
   MOV AH,0
   JMP OCT_ADDR1
OCT_ADDR2:
   MOV DH, AL
OCT_PRINT:
   POP DX                  ;pop  digit
   MOV DL,DH
   MOV DH,0
   ADD DX,30H ; ascii value
   MOV AH,2
   INT 21H
   LOOP OCT_PRINT
   POP BX
   PRINT '='
   RET
ENDP PRINT_OCT

PRINT_BIN PROC NEAR
   PUSH BX
   MOV AH,0
   MOV AL,BL
   MOV BL,2
```

```
    MOV CX,0
BIN_ADDR1:
    DIV BL
    INC CX
    PUSH AX
    CMP AL,0
    JE BIN_ADDR2
    MOV AH,0
    JMP BIN_ADDR1
BIN_ADDR2:
    MOV DH, AL
BIN_PRINT:
    POP DX
    MOV DL,DH
    MOV DH,0
    ADD DX,30H
    MOV AH,2
    INT 21H
    LOOP BIN_PRINT
    POP BX
    NEW_LINE
    RET
ENDP PRINT_BIN

PRINT_HEX PROC NEAR
    PUSH BX
    MOV AH,0
    MOV AL,BL
    MOV BL,16
    MOV CX,0
HEX_ADDR1:
    DIV BL
    INC CX
    PUSH AX
    CMP AL,0
    JE HEX_ADDR2
    MOV AH,0
    JMP HEX_ADDR1
HEX_ADDR2:
    POP DX
    MOV DL,DH
    MOV DH,0
    CMP DL,10
    JL  HEX_ADDR3
    ADD DX,37H
    JMP HEX_PRINT
HEX_ADDR3:
    ADD DX,30H
HEX_PRINT:
    MOV AH,2
    INT 21H
    LOOP HEX_ADDR2
    POP BX
    PRINT '='
    RET
ENDP PRINT_HEX

CODE ENDS
```

2η Άσκηση

```
INCLUDE MACROS.ASM
  .8086
  .MODEL SMALL
  .STACK 256

DATA SEGMENT
  TABLE DB 256 DUP(?)
  MIN db ?
  MAX db ?
DATA ENDS

CODE_SEG SEGMENT
  ASSUME CS:CODE_SEG, DS:DATA

MAIN PROC FAR
  MOV AX,DATA
  MOV DS,AX
  MOV AL,254
  MOV DI,255
  MOV [TABLE + DI],255 ; bazoume to 255 stin teleutaia thesi
  MOV DI,254
  MOV [TABLE + DI], 0
  MOV DI,0

MAKE_TABLE:
  MOV [TABLE + DI],AL   ; bale ton arithmo ston pinaka
  INC DI
  DEC AL
  JNZ MAKE_TABLE
  MOV DI,0
  MOV AH,0
  MOV DX,0

MESOS_OROS:
  MOV AL,[TABLE + DI]
  ADD DX,AX       ; prosthetoume olous tous zugous kai tous apothikeumoume ston DX
  ADD DI,2        ; DI+=2, afou theloume mono zugous
  CMP AL,255
  JNE MESOS_OROS  ; O teleutaios zugos einai o 255. Ara oso den einai 25, loop
  MOV AX,DX       ; sum ston accumulator
  MOV BH,0
  MOV BL,128
  DIV BL          ; AX/128 gia na bro m.o.
  MOV AH,0
  CALL PRINT_HEX ; tipono ton m.o. se hex
  NEW_LINE
  MOV DI,0
  MOV MIN,255    ;arxikopoioume to elaxisto sto megisto dunato kai to megisto sto elaxisto
  MOV MAX,0
MIN_LABEL:
  MOV AL,[TABLE + DI]             ;load number of TABLE[DI]
  CMP MIN,AL
  JB MAX_LABEL   ; sugkrine me to min, ki an einai megalutero, prosperase to
  MOV MIN,AL  ; allios AL = new min
MAX_LABEL:
  CMP MAX,AL   ; sugkrine me to max, ki an einai mikrotero, prosperase
  JA LOOP_AGAIN
  MOV MAX,AL   ; allios max = al
LOOP_AGAIN:
  INC DI                      ;increase index
  CMP DI,255                ; an di = 256, telos
```

```
    JNE MIN_LABEL
    MOV AH,0
    MOV AL,MIN
    CALL PRINT_HEX
    PRINT ' '
    MOV AH,0
    MOV AL,MAX
    CALL PRINT_HEX
    NEW_LINE
ENDP
EXIT

PRINT_HEX PROC NEAR
    PUSH BX
    MOV AH,0
    MOV AL,BL
    MOV BL,16
    MOV CX,0
HEX_ADDR1:
    DIV BL
    INC CX
    PUSH AX
    CMP AL,0
    JE HEX_ADDR2
    MOV AH,0
    JMP HEX_ADDR1
HEX_ADDR2:
    POP DX
    MOV DL,DH
    MOV DH,0
    CMP DL,10
    JL  HEX_ADDR3
    ADD DX,37H
    JMP HEX_PRINT
HEX_ADDR3:
    ADD DX,30H
HEX_PRINT:
    MOV AH,2
    INT 21H
    LOOP HEX_ADDR2
    POP BX
    PRINT '='
    RET
ENDP PRINT_HEX
```

3η Άσκηση

```
INCLUDE MACROS.ASM
   .8086
   .MODEL SMALL
   .STACK 256

DATA SEGMENT
DATA ENDS

CODE SEGMENT
   ASSUME CS:CODE, DS:DATA

MAIN PROC FAR
   PRINT "X"
   PRINT "="
   CALL HEX_KEYB
   MOV DL,AL
   MOV BL,BH
   CALL HEX_KEYB
   MOV DH,AL

   PUSH DX
   PRINT BL
   PRINT BH
   POP DX

   ROL DL,4
   ADD DL,DH

   PUSH DX
   PRINT ' '
   PRINT 'Y'
   PRINT '='
   CALL HEX_KEYB
   MOV DL,AL
   MOV BL,BH
   CALL HEX_KEYB
   MOV DH,AL

   PUSH DX
   PRINT BL
   PRINT BH
   POP DX

   MOV BL,DL
   MOV BH,DH
   ROL BL,4
   ADD BL,BH

   POP DX
   NEW_LINE

   PUSH BX
   PUSH DX
   PRINT 'X'
   PRINT '+'
   PRINT 'Y'
   PRINT '='
   POP DX

   PUSH DX    ; prostheto kai tupono to apotelesma
   MOV DH,0H
```

```
   MOV BH,0H
   ADD DX,BX
   PUSH AX
   MOV AX, DX
   CALL PRINT_DECIMAL
   POP AX
   POP DX
   POP BX

   PUSH DX
   PRINT ' '
   PRINT 'X'
   PRINT '-'
   PRINT 'Y'
   PRINT '='
   POP DX

   PUSH BX
   PUSH DX
   MOV DH,0H
   MOV BH,0H
   CMP DL,BL       ; tsekare an thetiko i arnitiko apotelesma
   JAE THETIKO
   PUSH DX
   PRINT '-'
   POP DX
   SUB BL,DL
   MOV DL,BL
   JMP RESULT
THETIKO:
   SUB DL,BL
RESULT:
   MOV AX,DX
   CALL PRINT_DECIMAL
   POP DX
   POP BX

   RET
MAIN ENDP

; Oles oi routines einai idies me autes tou bibliou
; opote den sumperilabame sxolia

HEX_KEYB PROC NEAR
   PUSH DX
IGNORE:
   READ
   CMP AL,'Q'
   JE ADDR2
   CMP AL,30H
   JL IGNORE
   CMP AL,39H
   JG ADDR1
   PUSH AX
   PRINT AL
   POP AX
   SUB AL,30H
   JMP ADDR2
ADDR1:
   CMP AL,'A'
   JL IGNORE
   CMP AL,'F'
   JG IGNORE
```

```
   PUSH AX
   PRINT AL
   POP AX
   SUB AL,37H
ADDR2:
   POP DX
   RET
HEX_KEYB ENDP

PRINT_HEX PROC NEAR
   CMP DL,9
   JLE ADDR3
   ADD DL,37H
   JMP ADDR4
ADDR3:
   ADD DL,30H
ADDR4:
   PRINT DL
   RET
PRINT_HEX ENDP

PRINT_DECIMAL  PROC NEAR
   MOV BL,10
   MOV CX,1
LOOP_10:
   DIV BL
   PUSH AX
   CMP AL,0
   JE PRINT_DIGITS_10
   INC CX
   MOV AH,0
   JMP LOOP_10
PRINT_DIGITS_10:
   POP DX
   MOV DL,DH
   MOV DH,0
   ADD DX,30H
   MOV AH,2
   INT 21H
   LOOP PRINT_DIGITS_10
   RET
ENDP PRINT_DECIMAL

CODE ENDS
   END
```

4η Άσκηση

```
INCLUDE MACROS.ASM
   .8086
   .MODEL SMALL
   .STACK 256

DATA SEGMENT
   TABLE DB 16 DUP(?)
DATA ENDS

CODE SEGMENT
   ASSUME CS : CODE, DS : DATA

MAIN PROC FAR
   MOV AX,DATA
   MOV DS,AX

START:
   MOV DI,0
   MOV CL,0

READ:
   READ_IN
   CMP AL,13
   JE QUIT        ; enter = telos
   CMP AL,'0'
   JL READ        ; dexomaste mono arithmous metaksu 0-9
   CMP AL,'9'
   JNA OK
   CMP AL,'A'
   JL READ        ; dexomaste mono kefalaia grammata
   CMP AL,'Z'
   JG READ

OK:
   MOV [TABLE + DI],AL
   INC DI
   INC CL
   CMP CL,17  ; 16 chars + enter char
   JNZ READ

   MOV DI,0
   MOV CL,0

PRINT1: ; tupose oles tis theseis tou pinaka
   MOV AL,[TABLE + DI]
   PRINT AL
   INC DI
   INC CL
   CMP CL,15 ; an cl!=15, exo akoma chars na tuposo
   JNZ PRINT1

   NEW_LINE

   MOV DI,0
   MOV CL,0

PRINT2:
   MOV AL,[TABLE + DI]
   CMP AL,3AH ; an mikrotero you 3A hex, tote einai arithmos kai ton tuponoume
   JA CONTINUE     ; an einai megalutero, einai gramma kai to prospername
```

```
   PRINT AL

CONTINUE:
   INC DI
   INC CL
   CMP CL,15
   JB PRINT2
   ; telos arithmoi, ara tuponoume - kai pame sta grammata
   PRINT '-'
   MOV DI,0
   MOV CL,0

PRINT_LETTERS:
   MOV AL,[TABLE + DI]
   CMP AL,40H
   JBE NEXT_LETTER

   ADD AL,32 ; an > 40H, einai kefalaio gramma kai prosthetoume 23 dec gia na ginei mikro
   PRINT AL

NEXT_LETTER:
   INC DI
   INC CH
   CMP CH,16
   JB PRINT_LETTERS
   NEW_LINE ; telos, tuponoume new line
   JMP START    ; kai jmp start gia sunexi leitourgia

QUIT:
   EXIT
   MAIN END
```

5η Άσκηση

```
INCLUDE MACROS.ASM
   .8086
   .MODEL SMALL
   .STACK 256

DATA SEGMENT
   MSG1 DB 0AH,0DH,'START(Y,N):$'
   MSG2 DB 0AH,0DH,'ERROR$'
DATA ENDS

CODE SEGMENT
   ASSUME CS : CODE, DS : DATA

MAIN PROC FAR
   MOV AX, DATA
   MOV DS, AX
 START:
   PRINT_STR MSG1              ;PRINT STARTING MESSAGE
   CALL HEX_KEYB      ;INPUT FIRST DIGIT
   CMP AL,'Y'                  ;if Y start
   JZ INPUT2
   CMP AL,'N'                  ;if N end
   JZ ENDING
   JMP START                   ;ELSE AGAIN
INPUT2:
   MOV AX,0H
   CALL HEX_KEYB   ;INPUT FIRST DIGIT
   MOV BH,AL
   CMP AL,'Y'
   JE INPUT2                   ;IF Y START ALL OVER
   CMP AL,'N'
   JE ENDING                   ; IF N END
   CALL HEX_KEYB      ;INPUT SECOND DIGIT
   CMP AL,'Y'
   JE INPUT2
   CMP AL,'N'
   JE ENDING
   MOV BL,AL
   ROL BL,4
   CALL HEX_KEYB      ;INPUT THIRD DIGIT
   CMP AL,'Y'
   JE INPUT2
   CMP AL,'N'
   JE START
   ADD BL,AL
   CMP BX,500D                 ;IF <500 (ASSUMING>0) CASE 1
   JNA FIRST_CASE
   CMP BX,700D                 ;IF (500<)x<700 (ASSUMING>0) CASE 2
   JNA SECOND_CASE
   CMP BX,1000D    ;IF (700<)x<1000 (ASSUMING>0) CASE 3
   JNA THIRD_CASE
   PRINT_STR MSG2
   NEW_LINE
   JMP START
FIRST_CASE:
   MOV AX,BX
   MOV DX,50D
   DIV DX          ;FIND V IN FIRST CASE=T*10/500
   JMP ENDING
SECOND_CASE:
   MOV AX,BX
```

```
   SUB AX,500D
   MOV DL,80H
   MUL DL
   MOV DX,2000D
   DIV DL
   ADD AX,01H        ;FIND V IN SECOND CASE=1+0,8(T-500)*10/200
   JMP ENDING
THIRD_CASE:
   MOV AX,BX
   SUB AX,700D
   MOV DL,20H
   MUL DL
   MOV DX,300D
   DIV DL
   ADD AX,18D        ;FIND V IN THIRD CASE=1+0,2(T-700)*10/300
   MOV DL,10D
   DIV DL
   JMP ENDING

OUTPUT:
   MOV DX,4095D   ;APPLY A/C EQUATION (=4095*V/2)
   MUL DL
   MOV DL,2D
   DIV DL
   MOV BX,AX
   MOV DX,1000D
   DIV DL
   PRINT_DEC                ;PRINT D3,D2
   MUL DL
   MOV CX,AX
   MOV DL,10D
   MOV AX,BX
   SUB AX,CX
   MOV CX,AX
   DIV DL
   PRINT_DEC                ;PRINT D1,D0
   MOV CX,AX
   PRINT ','                ;PRINT ,
   SUB BX,CX
   MOV AX,BX
   PRINT_DEC                ;PRINT D(-1)
   JMP START
ENDING:
   EXIT
   MAIN ENDP


HEX_KEYB PROC NEAR
   PUSH DX
IGNORE:
   READ
   CMP AL,'Y'
   JE ADDR2
   CMP AL,'N'
   JE ADDR2
   CMP AL,30H
   JL IGNORE
   CMP AL,39H
   JG ADDR1
   SUB AL,30H
   JMP ADDR2
ADDR1:
   CMP AL,'A'
```

```
   JL IGNORE
   CMP AL,'F'
   JG IGNORE
   SUB AL,37H
ADDR2:
   POP DX
   RET
HEX_KEYB ENDP
```