



5^η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

Ασκήσεις Προσομοίωσης

1^η ΑΣΚΗΣΗ

```
READ MACRO          ; Μακροεντολή για διάβασμα χαρακτήρα
    MOV AH, 8
    INT 21H
ENDM
PRINT MACRO CHAR    ; Μακροεντολή για εκτύπωση χαρακτήρα
    MOV DL, CHAR
    MOV AH, 2
    INT 21H
ENDM
NEW_LINE MACRO      ; Μακροεντολή για δημιουργία νέας γραμμής
    MOV DL, 0AH
    MOV AH, 2
    INT 21H
    MOV DL, 0DH
    MOV AH, 2
    INT 21H
ENDM
CODE_SEG SEGMENT
    ASSUME CS:CODE_SEG

MAIN PROC FAR
START:
    CALL HEX_KEYB    ; Κληση της διαδικασιας HEX_KEYB
    MOV AH,00H
    MOV BX,AX
    MOV AX,10H
    MUL BX
    MOV BX,AX
    CALL HEX_KEYB    ; Κληση της διαδικασιας HEX_KEYB
    ADD BX,AX
    PRINT '='
    CALL PRINT_DEC    ; Κληση της διαδικασιας PRINT_DEC
    PRINT '='
    CALL PRINT_OCT    ; Κληση της διαδικασιας PRINT_OCT
    PRINT '='
    CALL PRINT_BIN    ; Κληση της διαδικασιας PRINT_BIN
    NEW_LINE
    JMP START
ENDING:
    MOV AX,4C00H
    INT 21H
MAIN ENDP
```

HEX_KEYB PROC NEAR ; Διαδικασία διαβάσματος δεκαεξαδικού αριθμού
PUSH DX

IGNORE:

READ
PRINT AL
CMP AL,'Q'
JE ENDING
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_DEC PROC NEAR ; Διαδικασία υπολογισμού & εκτύπωσης ενός 16αδικού αριθμού σε
; δεκαδική μορφή

PUSH CX
PUSH AX
PUSH BX
MOV BH,00H
MOV AX, BX
MOV CX,0

ADDR3:

MOV DX,0
MOV BX,0AH
DIV BX
PUSH DX
INC CX
CMP AX,0
JNE ADDR3

ADDR4:

POP DX
ADD DX,30H
PRINT DL
LOOP ADDR4
POP BX
POP AX
POP CX
RET

PRINT_DEC ENDP

PRINT_OCT PROC NEAR ; Διαδικασία υπολογισμού & εκτύπωσης ενός 16αδικού αριθμού σε
; οκταδική μορφή

PUSH CX
PUSH AX
MOV AX,BX
PUSH BX
MOV CX,0

ADDR5:

```

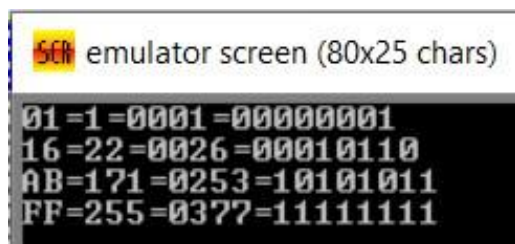
        MOV DX,0
        MOV BX,8
        DIV BX
        PUSH DX
        INC CX
        CMP AX,0
        JNZ ADDR5
        MOV BX,CX
ADDR6:
        MOV AX,BX
        CMP CX,BX
        JNE NOTFIRST
        PRINT '0'
        POP DX
        LOOP ADDR6
NOTFIRST:
        POP DX
        ADD DX,30H
        PRINT DL
        LOOP ADDR6
        POP BX
        POP AX
        POP CX
        RET
PRINT_OCT ENDP

PRINT_BIN PROC NEAR      ; Διαδικασία υπολογισμού & εκτύπωσης ενός 16αδικού αριθμού σε
                          ; δυαδική μορφή
        PUSH AX
        MOV AX,BX
        PUSH CX
        MOV CX,8
ADDR7:
        SHL BL,1
        MOV DL,0
        ADC DL,30H
        PRINT DL
        LOOP ADDR7
        POP CX
        MOV BX,AX
        POP AX
        RET
PRINT_BIN ENDP

CODE_SEG ENDS
END MAIN

```

Ενδεικτικό παράδειγμα εξόδου για εισαγωγή των δεκαεξαδικών 01, 16, AB, FF



2^η ΑΣΚΗΣΗ

```
DATA_SEG SEGMENT
    TABLE DB 1024 DUP(?)      ; Δημιουργία πίνακα 256*4 = 1024 θέσεων
    NEW_LINE DB 0AH,0DH,'$'
DATA_SEG ENDS
CODE_SEG SEGMENT
    ASSUME CS:CODE_SEG, DS:DATA_SEG

MAIN PROC FAR
    MOV AX,DATA_SEG
    MOV DS,AX
    MOV CX,0100H                ; Μετρητής CX = 256
    MOV BX,0000H                ; Καταχωρητής OFFSET
    MOV AL,00FEH                ; AX = 254 (ο πρώτος αριθμός που πρέπει να αποθηκευτεί)
    MOV AH,0000H
    MOV DL,TABLE
    MOV DH,0000H
    ADD BX,DX

ADR1: MOV [BX],AL                ; Αποθήκευση αριθμού στον πίνακα
    DEC AL
    ADD BX,0004H
    LOOP ADR1                    ; Επανάληψη για όσο CX > 0 (256 φορές)
    MOV CX,0000H                ; Μετρητής άρτιων αριθμών
    SUB BX,0004H
    MOV AX,0000H                ; Συσσωρευτής αθροίσματος

ADR2: ADD BX,DX
    TEST [BX],0001H              ; Έλεγχος άρτιου ή περιττού
    JNZ SKIP_ADD                 ; Αν είναι περιττός άλμα στην ετικέτα SKIP_ADD
    ADD AX,[BX]                  ; Αν είναι άρτιος προσθήκη στον AX
    INC CX

SKIP_ADD:
    SUB BX,0004H
    JNZ ADR2
    ADD AX,[BX]
    INC CX
    MOV DX,0000H
    DIV CX                       ; Διαίρεση του AX (άθροισμα άρτιων) με τον CX (μετρητής)
    CALL PRINT_HEX
    PRINT_STR NEW_LINE

MIN_MAX:
    ADD TABLE,1
    MOV CX,00FFH
    MOV BX,0000H                ; Μεγαλύτερος αριθμός (αρχικοποίηση με 0)
    MOV DX,0000H                ; Μικρότερος αριθμός (αρχικοποίηση με 0)
    MOV AX,0000H                ; Αρχικοποίηση με 0 του AX ώστε να αποθηκεύονται οι
                                ; αριθμοί από τον πίνακα

ADR3: MOV AL,[TABLE]
    CMP BX,AX                    ; Σύγκριση αριθμού με το BX, αν είναι μικρότερος άλμα στην
                                ; ετικέτα SKIP_MAX
    JG SKIP_MAX
    MOV BX,AX                    ; αλλιώς αποθήκευση του νέου μεγαλύτερου στον BX

SKIP_MAX:
    CMP DX,AX                    ; Σύγκριση αριθμού με το DX (0), αν είναι μεγαλύτερος
    JL SKIP_MIN                 ; άλμα στην ετικέτα SKIP_MIN
    MOV DX,AX                    ; αλλιώς αποθήκευση του νέου μικρότερου στον BX

SKIP_MIN:
    SUB TABLE,1
    LOOP ADR3
    MOV AX,BX
    CALL PRINT_HEX                ; Εκτύπωση μεγαλύτερου αριθμού
    MOV AX,DX
```

```

        CALL PRINT_HEX          ; Εκτύπωση μεγαλύτερου αριθμού
        EXIT
MAIN    ENDP

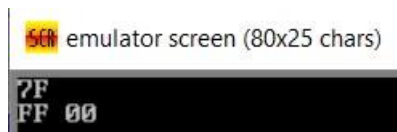
PRINT_STR MACRO STRING          ; Μακροεντολή για εκτύπωση string
        MOV DX, OFFSET STRING
        MOV AH,9
        INT 21H
ENDM
PRINT MACRO CHAR                ; Μακροεντολή για εκτύπωση χαρακτήρα
        MOV DL, CHAR
        MOV AH, 2
        INT 21H
ENDM
EXIT MACRO                      ; Μακροεντολή εξόδου
        MOV AX, 4C00H
        INT 21H
ENDM

PRINT_HEX PROC NEAR            ; Διαδικασία εκτύπωσης διψήφιου HEX
        PUSH DX
        MOV BL,AL
        AND AL,00F0H
        ROR AL,4
        CMP AL,9
        JG ITS_LET
        ADD AL,30H
        JMP MONADES
ITS_LET:
        ADD AL,37H
MONADES:
        PRINT AL
        MOV AL,BL
        AND AL,000FH
        CMP AL,9
        JG ITS_LET2
        ADD AL,30H
        PRINT AL
        JMP TELOS
ITS_LET2:
        ADD AL,37H
        PRINT AL
TELOS:
        PRINT ' '
        POP DX
        RET
PRINT_HEX ENDP

CODE_SEG ENDS
END MAIN

```

Η έξοδος του προγράμματος είναι η ακόλουθη:



Όπως ήταν αναμενόμενο ο μέσος όρος των άρτιων αριθμών είναι $7FH = 127$, η μέγιστη τιμή είναι $FFH = 255$, ενώ η ελάχιστη $00H = 0$.

3^η ΑΣΚΗΣΗ

```
READ MACRO
    MOV AH, 8
    INT 21H
ENDM
PRINT MACRO CHAR
    MOV DL, CHAR
    MOV AH, 2
    INT 21H
ENDM
CODE_SEG SEGMENT
    ASSUME CS:CODE_SEG

MAIN PROC FAR
    MOV BX,16
    MOV CX,16
    PRINT "X"
    PRINT "="
    CALL HEX_KEYB      ; Εισαγωγή στον AL του 1ου HEX ψηφίου
    MUL BL              ; Πολλαπλασιασμός του με 16
    MOV BL,AL           ; Αποθήκευση του αποτελέσματος στον BL
    CALL HEX_KEYB      ; Εισαγωγή στον AL του 2ου HEX ψηφίου
    ADD BL,AL           ; Άθροισμα των δύο HEX στον BL
    PRINT " "
    PRINT "Y"
    PRINT "="
    CALL HEX_KEYB      ; Εισαγωγή στον AL του 1ου HEX ψηφίου
    MUL CL              ; Πολλαπλασιασμός του με 16
    MOV CL,AL           ; Αποθήκευση του αποτελέσματος στον CL
    CALL HEX_KEYB      ; Εισαγωγή στον AL του 2ου HEX ψηφίου
    ADD CL,AL           ; Άθροισμα των δύο HEX στον CL
    PRINT 0AH
    PRINT 0DH
    PRINT "X"
    PRINT "+"
    PRINT "Y"
    PRINT "="
    MOV AX,CX
    ADD AX,BX           ; Άθροισμα των δύο HEX
    CALL PRINT_DEC      ; Εκτύπωση του αθροίσματος σε δεκαδική μορφή
    PRINT 0AH
    PRINT 0DH
    PRINT "X"
    PRINT "-"
    PRINT "Y"
    PRINT "="
    MOV AX,BX           ; Αποθήκευση του πρώτου HEX στον AX
    CMP AX,CX           ; Σύγκρισή του με τον δεύτερο HEX
    JG NOSIGN           ; Αν είναι μεγαλύτερος άλμα στην ετικέτα NOSIGN
    PUSH AX
    PRINT "-"           ; Αλλιώς η διαφορά θα είναι αρνητική
    POP AX
    SUB CX,AX           ; Αφαίρεση του 1ου HEX από τον 2ο
    MOV AX,CX
    CALL PRINT_DEC
    JMP FINISH
NOSIGN:
    SUB AX,CX           ; Αφαίρεση του 2ου HEX από τον 1ο
    CALL PRINT_DEC
    JMP FINISH
```

HEX_KEYB PROC NEAR ; ; Διαδικασία εισαγωγής HEX

PUSH DX ;

IGNORE:

READ

CMP AL,30H

JL IGNORE

CMP AL,39H

JG ADDR1

PRINT AL

SUB AL,30H

JMP ADDR2

ADDR1:

CMP AL,'A'

JL IGNORE

CMP AL,'F'

JG IGNORE

PRINT AL

SUB AL,37H

ADDR2:

POP DX

RET

HEX_KEYB ENDP

PRINT_DEC PROC NEAR ; ; Διαδικασία μετατροπής HEX σε δεκαδική μορφή και εκτύπωση

PUSH CX

PUSH AX

PUSH BX

MOV BH,00H

MOV CX,0

ADDR3:

MOV DX,0

MOV BX,0AH

DIV BX

PUSH DX

INC CX

CMP AX,0

JNE ADDR3

ADDR4:

POP DX

ADD DX,30H

PRINT DL

LOOP ADDR4

POP BX

POP AX

POP CX

RET

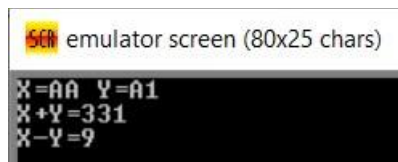
PRINT_DEC ENDP

FINISH:

CODE_SEG ENDS

END MAIN

Ενδεικτική έξοδος του προγράμματος για την εισαγωγή των δεκαεξαδικών AAH και A1H



4^η ΑΣΚΗΣΗ

```
PRINT MACRO CHAR                ; Μακροεντολή εκτύπωσης χαρακτήρα
    PUSH DX
    PUSH AX
    MOV DL,CHAR
    MOV AH,2
    INT 21H
    POP AX
    POP DX
ENDM
EXIT MACRO                      ; Μακροεντολή εξόδου
    PUSH AX
    MOV AX,4C00H
    INT 21H
    POP AX
ENDM
READ MACRO                     ; Μακροεντολή εισαγωγής χαρακτήρα
    MOV AH,08H
    INT 21H
ENDM

DATA_SEG SEGMENT
TABLE 16 DUP(?)
DATA_SEG ENDS

STACK_SEG SEGMENT
DB 16 DUP(?)                  ; Στοιβά 16 στοιχείων
STACK_SEG ENDS

CODE_SEG SEGMENT
ASSUME CS:CODE_SEG,SS:STACK_SEG ,DS:DATA_SEG
MOV AX,DATA_SEG
MOV DS,AX

MAIN PROC FAR
LOOP_1:
    MOV AL,0DH
    PRINT AL
    MOV BP,0
NEXT_1:
    READ
    CMP AL,0DH                ; Τερματισμός με τον χαρακτήρα ENTER
    JE FINISH
    CMP AL,30H                ; Έλεγχος για αποδεκτό χαρακτήρα
    JL NEXT_1                 ; Αν δεν είναι αποδεκτός άλμα στην ετικέτα NEXT_1
    CMP AL,3AH                ; Αν είναι αποδεκτός, έλεγχος αν πρόκειται για αριθμό
    JL NUMORLET               ; Αν είναι αριθμός άλμα στην ετικέτα NUMORLET
    CMP AL,40H                ; αλλιώς ξανά έλεγχος για αποδεκτό χαρακτήρα
    JLE NEXT_1                ; Αν δεν είναι αποδεκτός άλμα στην ετικέτα NEXT_1
    CMP AL,5BH ;A-Z           ; Αν είναι αποδεκτός, έλεγχος αν πρόκειται για κεφαλαίο γράμμα
    JL NUMORLET               ; Αν είναι κεφαλαίο γράμμα άλμα στην ετικέτα NUMORLET
    JMP NEXT_1                ; αλλιώς άλμα στην ετικέτα NEXT_1
NUMORLET:
    PRINT AL
    MOV [BP],AL                ; Αποθήκευση του χαρακτήρα στον πίνακα
    INC BP
    CMP BP,16
    JNE NEXT_1                 ; Επανάληψη μέχρι να γεμίσει ο πίνακας
    MOV AL,0AH
    PRINT AL
```




```

        MOV AL, 0DH
        PRINT AL
        MOV BP,0
PRINT_NUMBERS:                ; Πέρασμα από τον πίνακα και εκτύπωση όλων των αριθμών
        MOV AL,[BP]
        CMP AL,3AH
        JL PRINT_NUM
        JMP BACKN
PRINT_NUM:
        PRINT AL
BACKN:
        INC BP
        CMP BP,16
        JNE PRINT_NUMBERS
        MOV AL,2DH
        PRINT AL
        MOV BP,0
PRINT_LETTERS:                ; Πέρασμα από τον πίνακα και εκτύπωση των αντίστοιχων πεζών
                                ; γραμμάτων
        MOV AL,[BP]
        ADD AL,20H
        CMP AL,60H
        JG PRINT_LET
        JMP BACKL
PRINT_LET:
        PRINT AL
BACKL:
        INC BP
        CMP BP,16
        JNE PRINT_LETTERS
        MOV AL,0AH
        PRINT AL
        JMP LOOP_1
FINISH:
        EXIT
CODE_SEG ENDS
MAIN ENDP

```

Ενδεικτική έξοδος του προγράμματος:

 emulator screen (80x25 chars)

```

ABCDEF1234567890
1234567890-abcdef
1A2B3C4D5E6F1A2B
12345612-abcdefab
_

```

5" ΑΣΚΗΣΗ

```
READ MACRO                                ; Μακροεντολή για εισαγωγή χαρακτήρα
    MOV AH, 8
    INT 21H
ENDM
PRINT MACRO CHAR                          ; Μακροεντολή για εκτύπωση χαρακτήρα
    MOV DL, CHAR
    MOV AH, 2
    INT 21H
ENDM
PRINT_STR MACRO STRING                    ; Μακροεντολή για εκτύπωση string
    MOV DX, OFFSET STRING
    MOV AH,9
    INT 21H
ENDM
EXIT MACRO                                ; Μακροεντολή εξόδου
    MOV AX, 4C00H
    INT 21H
ENDM
PRINT_DEC MACRO                           ; Μακροεντολή για εκτύπωση δεκαδικού αριθμού
    ADD DL, 30H
    MOV AH,2
    INT 21H
ENDM

DATA SEGMENT
    BEGIN DB 'START $'
    NEW DB '(Y,N):$'
    ERROR DB 'ERROR$'
    DOT DB '.$'
    NEWLINE DB 0AH,0DH,'$'
    TABLE DB 1 DUP(?)
ENDS

STACK SEGMENT
    DW 128 DUP(0)
ENDS

CODE SEGMENT
ENDS

MAIN PROC FAR
    MOV AX, DATA
    MOV DS, AX
    PRINT_STR BEGIN
START:
    PRINT_STR NEW
    CALL HEX_KEYB                          ; Εισαγωγή 1ο HEX
    CMP AL,'N'                             ; Τερματισμός σε περίπτωση εισαγωγής του 'N'
    JE THEEND
    MOV BL,AL
    ROL BX,4                               ; Αποθήκευση του στα 4 MSBs του BX
    CALL HEX_KEYB                          ; Εισαγωγή 2ο HEX
    CMP AL,'N'
    JE THEEND
    OR BL,AL                               ; Αποθήκευση του στα 4 LSBs του BX
    ROL BX,4
    CALL HEX_KEYB                          ; Εισαγωγή 3ο HEX
    CMP AL,'N'
    JE THEEND
```

```

    OR BL,AL
    CMP BX,2047          ; 1η περίπτωση A/DE (0-2047)
    JLE FIRST
    CMP BX,3685          ; 2η περίπτωση A/DE (2047-3685)
    JLE SECOND
    CMP BX,4095          ; 3η περίπτωση A/DE (3685-4095)
    JL THIRD
    PRINT_STR ERROR
    PRINT_STR NEWLINE
    JMP START
FIRST:
    MOV DX,1000
    MOV AX,BX
    MUL DX                ; 1000*A/D
    MOV CX,4095
    DIV CX                ; (1000*A/D)/4095
    MOV [TABLE],AH
    MOV BX,10
    MUL BX
    JMP CONV
SECOND:
    MOV DX,400
    MOV AX,BX
    MUL DX
    MOV CX,3276
    DIV CX
    ADD AX,250            ; (400*A/D/3276)+250
    MOV [TABLE],AH
    MOV BX,10
    MUL BX
    JMP CONV
THIRD:
    MOV DX,600
    MOV AX,BX
    MUL DX
    MOV CX,819
    DIV CX
    SUB AX,2000           ; (600*A/D/819)-2000
    MOV [TABLE],AH
    MOV BX,10
    MUL BX
CONV:
    MOV CX,0
DIGIT:
    MOV DX,0
    MOV BX,10
    DIV BX
    PUSH DX
    INC CX
    CMP AX,0
    JNE DIGIT
    DEC CX
TIGID:
    POP DX
    PRINT_DEC
    LOOP TIGID
    PRINT '.'
    POP DX
    MOV DL,[TABLE]
    PRINT_DEC
    PRINT_STR NEWLINE

```

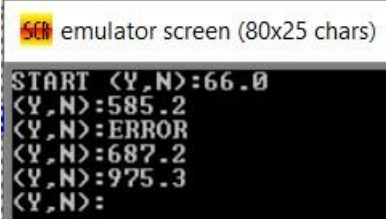
```

        JMP START
THEEND:
        EXIT
MAIN ENDP

HEX_KEYB PROC NEAR
        PUSH DX
IGNORE:
        READ
        CMP AL,'N'
        JE THEEND
        CMP AL,30H
        JL IGNORE
        CMP AL,39H
        JG ADDRESS1
        SUB AL,30H
        JMP ADDRESS2
ADDRESS1:
        CMP AL,'A'
        JL IGNORE
        CMP AL,'F'
        JG IGNORE
        SUB AL,37H
ADDRESS2:
        POP DX
        RET
HEX_KEYB ENDP

```

Ενδεικτική έξοδος του προγράμματος για την εισαγωγή των δεκαεξαδικών 111H, ABCH, FFFH, DF3H και FDDH:



```

5ch emulator screen (80x25 chars)
START <Y,N>:66.0
<Y,N>:585.2
<Y,N>:ERROR
<Y,N>:687.2
<Y,N>:975.3
<Y,N>:

```