Εργαστήριο Μικροϋπολογιστών

3η Εργαστηριακή Άσκηση

Ομάδα Γ04

Συνεργάτες:

- **Σκουλός Ραφαήλ, Α.Μ : 03112404**
- **Αναστάσης Σταθόπουλος, Α.Μ : 03112140**
- **Τζίνης Ευθύμιος , Α.Μ : 03112007**

1ο Θέμα

Ακολουθεί ο κώδικας με σχόλια:

```
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


DATA SEGMENT
      MSG1 DB 0AH, 0DH, "GIVE AN 8-BIT BINARY NUMBER: $"
      MSG2 DB 0AH, 0DH, "DECIMAL: $"
ENDS


STACK SEGMENT
      DW 128 DUP<0>
ENDS



CODE SEGMENT
      ASSUME CS:CODE, DS:DATA, SS:STACK, ES:DATA

START:
      MOV AX,DATA
      MOV DS,AX
      MOV ES,AX

      PRINT_STR MSG1

      MOV BL,00H         ;SAVE BINARY NUMBER
      MOV CX,08H         ;COUNTER FOR 8 LOOPS
```

```
IGNORE:
      READ
      CMP AL,51H        ;ASCII CODE FOR 'Q'
      JE FINISH
      CMP AL,30H        ;IF LESS THAN '0' IGNORE
      JL IGNORE
      CMP AL,31H        ;IF MORE THAN '1' IGNORE
      JG IGNORE
      SAL BL,01H
      PUSH AX           ;SAVE AL BECAUSE COMMAND <PRINT>
EFFECTS IT
      PRINT AL
      POP AX
      SUB AL,30H        ;CONVERT ASCII CODE TO NUMBER
      ADD BL,AL         ;HOLD INPUT AT BL
      LOOP IGNORE

      PRINT_STR MSG2

      MOV CH,00H
      MOV CL,00H

COUNT_HUN:              ;COUNT HUNDREDS
    CMP BL,64H
    JNA COUNT_DEC       ;CHECK IF BL IS NOT ABOVE 100 BECAUSE I
                        ;WANT CHECK CF ALSO
    INC CL
    SUB BL,64H
    JMP COUNT_HUN

COUNT_DEC:              ;COUNT DECADES
    CMP BL,0AH
    JL SHOW
```

```
    INC CH
    SUB BL,0AH
    JMP COUNT_DEC

SHOW:                       ;CONVERT RESULT TO ASCII CODE
     ADD CL,30H             ;PRINT IT
     PRINT CL
     ADD CH,30H
     PRINT CH
     ADD BL,30H
     PRINT BL

     JMP START

FINISH:
     ENDS
```

2ο Θέμα

Ακολουθεί ο κώδικας με σχόλια:

```
DATA_SEG    SEGMENT

    NUMBERS 100 DUP(0)
    MS1 DB  0AH,0DH,"GIVE DECIMAL DIGITS :$"
    MS2 DB  0AH,0DH,"HEX=$"
    MS3 DB  0AH,0DH,"THE MAXIMUM NUMBER OF DIGITS MUST BE 100
$"
INDEX1 DB 0      ;POINTER TO THE FIRST NON ZERO ELEMENT OF THE
                 ;ARRAY NUMBERS
```

```
    INDEX2 DB 0      ;POINTER TO THE LAST ELEMENT OF THE ARRAY
                     ;NUMBERS
    CHECK DB 0
    CHECK2 DB 0
    COUNTER DB 0
    COUNTER2 DB 0
DATA_SEG    ENDS

CODE_SEG SEGMENT
    ASSUME CS:CODE_SEG,DS:DATA_SEG


PRINT_STR MACRO STRING    ;MACRO TO PRINT A STRING
    MOV DX,OFFSET STRING
    MOV AH,9
    INT 21H

ENDM

PRINT MACRO CHAR       ;MACRO TO PRINT A CHAR
    MOV DL,CHAR
    MOV AH,2H
    INT 21H
ENDM


PRINT_HEX PROC NEAR
    CMP DL,9
    JLE ADDR3
    ADD DL,37H
    JMP ADDR4
ADDR3:
```

```
    ADD DL,30H
ADDR4:
    MOV AH,02H
    INT 21H
    RET
PRINT_HEX ENDP


READ_NUM PROC NEAR      ;PROC TO READ A NUMBER
     PUSH DX

IGNORE:
    MOV AH,08H      ;READ THE NUMBER
    INT 21H
    MOV CL,INDEX2
    CMP CL,4        ;CHECK IF I HAVE ALREADY READ 4 NUMBERS
    JL OXI_TELOS
    CMP AL,0DH      ;IF SO CHECK FOR ENTER BUTTON
    JE TELOS
OXI_TELOS:
    CMP AL,30H      ;CHECK IF THE CHARACTER IS A NUMBER
                    ;BETWEEN 0 AND 9
    JL IGNORE
    CMP AL,39H      ;IF NOT READ AGAIN
    JG IGNORE

    JMP KEEP_READING

TELOS:
    MOV CHECK,1     ;IF ENTER BUTTON IS PUSHED THE CHECK=1

KEEP_READING:
```

```
    POP DX
    RET
READ_NUM ENDP




MAIN PROC FAR
    MOV AX,DATA_SEG
    MOV DS,AX

START:
    PRINT_STR MS3
    PRINT_STR MS1
    MOV BX,OFFSET NUMBERS   ;INITIALIZE BX TO THE FIRST
                            ;ELEMENT OF ARRAY
    MOV CHECK,0
    MOV INDEX1,0
    MOV INDEX2,0
    MOV COUNTER,0
    MOV CHECK2,0

CREATE_ARRAY:
     CALL READ_NUM   ;READ NUMBER
     CMP CHECK,1     ;CHECK IF ENTER BUTTON IS PUSHED
     JE ENDOF_ARRAY ;IF SO  STOP READING
     MOV [BX],AL     ;ELSE PUT THE NUMBER YOU READ TO THE
ARRAY
     INC INDEX2      ;INCREASE INDEX2
     INC BX          ;GET TO THE NEXT CELL OF THE ARRAY
     JMP CREATE_ARRAY
```

```
ENDOF_ARRAY:
     MOV BX,OFFSET NUMBERS


FIND_NONZERO_ELEMENT:           ;FINDS THE PLACE OF THE
                                ;FIRST NON ZERO ELEMENT
     MOV CL,INDEX1
     MOV CH,INDEX2
     CMP CL,CH
     JG ENDOF_0
     MOV AL,[BX]
     SUB AL,30H    ;GET THE ASCII CODE OF THE ELEMENT
     CMP AL,0      ;IF I FIND ONE NON ZERO ELEMENT I'M DONE
     JNE ENDOF_0
     INC BX
     INC INDEX1
     JMP FIND_NONZERO_ELEMENT

ENDOF_0:
    MOV BX,OFFSET NUMBERS
    MOV CH,0
    MOV CL,INDEX2   ;CX=THE PLACE OF THE LAST ELEMENT OF ARRAY
    MOV AL,CL
    SUB AL,INDEX1   ;COUNTER2=THE NUMBER OF THE ELEMENTS LEFT
                    ;AFTER THE FIRST NON-ZERO
    MOV COUNTER2,AL

PRINT_NUMS:

    CMP CHECK2,1      ;IF WE HAVE NOT FOUND THE FIRST NON ZERO
                      ;ELEMENT WE DONT HAVE TO PUT COMMA
    JNE NOT_COMMA
```

```
    MOV AL, COUNTER2     ;ELSE WE CHECK IF THE NON ZERO
                         ;ELEMENTS LEFT ARE MULTIPLE OF 3
    MOV AH,0
    MOV DH,3
    DIV DH
    CMP AH,0

    JNE NOT_COMMA        ;IF SO WE PUT COMMA

    MOV DL,2CH
    MOV AH,2
    INT 21H

NOT_COMMA:

    MOV DH,[BX]
    PUSH AX              ;THEN WE PRINT THE NEXT ELEMENT OF THE
                         ;ARRAY

    PRINT DH
    POP AX
    SUB DH,30H
    CMP CHECK2,1         ;WE CHECK IF CKECK2=1
    JE DECRONLY          ;IF SO WE DECREASE THE ELEMENTS LEFT
    CMP DH,0             ;IF NOT  WE CHECK IF THIS ELEMENT IS 0
    JE DONT_DECR         ;IF SO WE DO NOTHING
    MOV CHECK2,1         ;ELSE WE MAKE CHECK2=1

DECRONLY:

   DEC COUNTER2          ;AND WE DECREASE THE COUNTER
```

```
DONT_DECR:

    INC BX
    LOOP PRINT_NUMS

INTO_HEX:
     MOV BX,OFFSET NUMBERS
     MOV CL,INDEX2
     MOV CH,0
     ADD BX,CX              ;CX=THE PLACE OF THE LAST ELEMENT
                            ;OF THE ARRAY


    SUB BX,4

    MOV DL,[BX]
    SUB DL,30H             ;WE CHECK IF THE LAST 4 NUMBERS WE
                           ;GOT ARE 307
    CMP DL,3               ;IF SO WE QUIT
    JNE OK

    INC BX

    MOV DL,[BX]
    SUB DL,30H
    CMP DL,0
    JNE OK

    INC BX                 ;IF NOT WE CONTINUE

    MOV DL,[BX]
```

```
    SUB DL,30H
    CMP DL,7
    JNE OK

    INC BX

    MOV DL,[BX]
    SUB DL,30H
    CMP DL,6
    JE QUIT


OK:
        MOV BX,OFFSET NUMBERS
        MOV CL,INDEX2
        MOV CH,0
        ADD BX,CX              ;CX=THE PLACE OF THE LAST
                               ;ELEMENT OF THE ARRAY


       SUB BX,4
       PRINT_STR MS2
       MOV CX,0
       MOV DX,0
       MOV INDEX2,4    ;WE USE INDEX2 AS A COUNTER AS WE DON'T
                       ;NEED IT ANYMORE

LOOPA:
     MOV DL,[BX]
     SUB DL,30H
     PUSH DX
     MOV AX,10           ;I MAKE THE 4 DIGIT NUMBER FROM THE 4
```

```
                        ;LAST ELEMENTS OF THE ARRAY
      MUL CX
      MOV CX,AX
      POP DX
      ADD CX,DX

      INC BX
      DEC INDEX2
      CMP INDEX2,0
      JG LOOPA

      MOV BX,CX
      MOV CX,4

ADDR2:

     ROL BX,4              ;WE USE 4 RIGHT SLIDES TO TAKE EACH
                           ;ONE OF THE NUMBER'S DIGITS
     MOV DX,BX
     AND DX,000FH
     CALL PRINT_HEX
     LOOP ADDR2
     JMP START

QUIT:
     MOV AX,4C00H          ;RETURN THE CONTROL TO THE OPERATING
                           ;SYSTEM
     INT 21H

CODE_SEG ENDS
   END MAIN
```

3ο Θέμα

Ακολουθεί ο κώδικας με σχόλια:

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

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

PRINT MACRO CHAR
    MOV DL,CHAR
    MOV AH,2H
    INT 21H
ENDM

PRINT_MESSAGE MACRO MESSAGE
    PRINT_STR MESSAGE
    PRINT_STR NEWLINE
ENDM

PRINT_ARRAY MACRO ARRAY, INDEXR
    LOCAL CHECKNEXT, EXIT_PRINT_ARRAY
    MOV BX,OFFSET ARRAY
    MOV CX,00H
    MOV CL,INDEXR             ;CL=COUNTER OF STORE ARRAY
    CMP CL,00H                ;IF CL IS ZERO THEN YOU DONT
```

```
                                ;HAVE TO PRINT ANYTHING
    JZ EXIT_PRINT_ARRAY

CHECKNEXT:
    MOV DL,[BX]
    PRINT DL                    ;PRINT THE CHECKING CELL
    INC BX                      ;SHOW TO THE NEXT CELL
    LOOP CHECKNEXT              ;DO THAT FOR THE WHOLE ARRAY
EXIT_PRINT_ARRAY:
ENDM

UPDATE  MACRO ARRAY, INDEXR     ;IN THIS MACRO WE SUPPOSE THAT
                                ;AL HAS THE VALUE WE WANT TO

        MOV BX,OFFSET ARRAY     ;INITIALIZE IT TO THE FIRST
                                ;ELEMENT OF ARRAY
        MOV DL,INDEXR           ;THE INDEX OF THE ARRAY
        MOV DH,00H
        ADD BX,DX               ;BX IS IN THE NEXT POSITION WE
                                ;WANT TO STORE
        MOV [BX],AL             ;OF COURSE AL HAS THE VALID
                                ;INPUT NUMBER OR CHARACTER
        INC INDEXR              ;COUNTER ++

ENDM

DATA SEGMENT
    BUFFER  DB "01234567890123"
    NUMBERS DB "01234567890123"  ;BUFFER HAS 14 CELLS FOR
                                 ;EVERY POSSIBLE INPUT
    LOWERCASE DB "aaaaaaaaaaaaaa"
    UPPERCASE DB "AAAAAAAAAAAAAA"
```

```
    INDEXN DB 0
    INDEXL DB 0
    INDEXU DB 0
    INDEXB DB 0
    MAX1   DB 0
    MAX2   DB 0
    WELC DB "PLEASE INSERT 14 CHARACTERS a-z OR A-Z OR ANY
DECIMAL DIGITS",0AH,0DH,'$'
    MSG1 DB "~YOU INSERTED~ $"
    MSG2 DB "~LOWERCASE DIGITS UPPERCASE~ $"
    MSG3 DB "~MAX1 MAX2~ $"
    NO_DIG DB "~YOU HAVE NOT ENTERED ANY DIGITS!~ $"
    ONLY_ONE DB "~YOU HAVE ENTERED ONLY ONE DIGIT SO ITS THE
MAX AS WELL~ $"
    BYE DB "ADIOS AMIGOS $"
    NEWLINE DB 0AH,0DH,'$'
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA

MAIN PROC FAR

        MOV AX,DATA
        MOV DS,AX

START:  PRINT_STR WELC
        MOV INDEXB,00H  ;WE WANT THE LOOP TO HAPPEN 14 DEC
TIMES
        MOV INDEXN,00H
        MOV INDEXU,00H ;INDEXES INITIALIZATION
        MOV INDEXL,00H
        MOV MAX1,30H
        MOV MAX2,30H    ;INITIALIZE THE MAXIMUM VALUES IN ZERO
```

```
                        ;(30H IN ASCII)

LOOPA:  READ            ;READ A CHAR  WE STORE IT IN AL
        CMP AL,'='      ;CHECK IF WE HAVE PRESSED '=' THAT MEANS
                        ;TERMINATE THE PROGRAM
        JZ TERMINATE
        CMP AL,0DH      ;CHECK IF WE HAVE PRESSED ENTER
        JZ ENTER

        CMP AL,'0'
        JL LOOPA        ;IF IT HAS ASCII CODE < 0 ITS NOT VALID
        CMP AL,'9'
        JLE DIGIT       ;IF IT IS BETWEEN 0 AND 9 ITS A DIGIT

        CMP AL,'A'      ;IF ITS NOT A NUMBER AND ITS CODE <
                        ;CODE(A) THEN NOT VALID
        JL LOOPA
        CMP AL,'Z'      ;IF ITS BETWEEN A , Z ITS A VALID
                        ;UPPERCASE LETTER THEN SAVE IT
        JLE UPPERC

        CMP AL,'a'      ;IF ITS NOT A NUMBER AND ITS CODE <
                        ;CODE(a) THEN NOT VALID
        JL LOOPA
        CMP AL,'z'      ;IF ITS BETWEEN a , z ITS A VALID
                        ;LOWERCASE LETTER THEN SAVE IT
        JG LOOPA        ;ELSE IGNORE IT
        JMP LOWERC

DIGIT:
FIRST_MAX:              ;INPUT NUMBER IS IN AL
        CMP AL,MAX1
```

```
        JL CONTINUE
        MOV MAX1,AL
CONTINUE:
        UPDATE NUMBERS, INDEXN
        JMP SAVE_IT

UPPERC: UPDATE UPPERCASE, INDEXU
        JMP SAVE_IT

LOWERC: UPDATE LOWERCASE, INDEXL

SAVE_IT:UPDATE BUFFER, INDEXB
        CMP INDEXB,0DH          ;WE WILL SAVE IT ONLY 14 DEC
                                ;TIMES THEN WE OUTPUT
        JZ WAIT_FOR_IT          ;IF WE HAVE 14 CHARACTERS OR
                               ;NUMBER THEN WAIT FOR ENTER TO
                               ;SHOW IT
        JMP LOOPA               ;ALL THE BUFFER INPUT ANYWAY

WAIT_FOR_IT:
        READ        ;READ A CHAR  WE STORE IT IN AL
        CMP AL,'='  ;CHECK IF WE HAVE PRESSED '=' THAT MEANS
                    ;TERMINATE THE PROGRAM
        JZ TERMINATE
        CMP AL,0DH  ;CHECK IF WE HAVE PRESSED ENTER
        JNZ WAIT_FOR_IT


ENTER:  PRINT_MESSAGE MSG1
        PRINT_ARRAY BUFFER, INDEXB   ;CALL THE MACRO TO SHOW
                                     ;ALL THE INPUT CHARS OR
                                     ;DIGITS
        PRINT_STR NEWLINE
```

```
        PRINT_MESSAGE MSG2
        PRINT_ARRAY LOWERCASE, INDEXL ;SHOW ONLY THE LOWERCASE
                                      ;CHARACTERS THAT YOU
                                      ;HAVE ENTERED
        CMP INDEXL,00H
        JZ CONTROL1                   ;DONT OUTPUT ANY SPACES
                                      ;IF WE DONT HAVE
                                      ;LOWERCASE CHARS
        PRINT ' '
CONTROL1:
        PRINT_ARRAY NUMBERS, INDEXN   ;ONLY THE NUMBERS
        CMP INDEXN,00H
        JZ CONTROL2                   ;DONT OUTPUT ANY SPACES
                                      ;IF WE DONT HAVE DIGITS
        PRINT ' '
CONTROL2:
        PRINT_ARRAY UPPERCASE, INDEXU ;ONLY THE UPPERCASE
                                      ;CHARACTERS
        PRINT_STR NEWLINE

FIND_THE_MAX:

        CMP INDEXN, 00H
        JNZ HAS_1
        PRINT_MESSAGE NO_DIG
        JMP EXIT

HAS_1:  CMP INDEXN, 01H
        JNZ HAS_2_OR_MORE
        PRINT_MESSAGE ONLY_ONE
        PRINT MAX1
        JMP FOUND_MAX
```

```
HAS_2_OR_MORE:

DO_MAX1_ZERO:
        MOV BX,OFFSET NUMBERS
        MOV CH,00H
        MOV CL,INDEXN             ;CL=COUNTER OF NUMBERS ARRAY

NEXT:
        MOV DL,[BX]
        CMP DL,MAX1
        JNZ CONT                 ;IF WE DID NOT FIND MAX1 THEN
                                  ;CHECK THE NEXT DIGIT
        MOV [BX],30H             ;ELSE DO IT ZERO (30H)
        JMP EXIT_DO_MAX1_ZERO
CONT:   INC BX                       ;SHOW TO THE NEXT CELL
        LOOP NEXT               ;DO THAT FOR THE WHOLE ARRAY
EXIT_DO_MAX1_ZERO:

FIND_MAX2:  ;NOW WE HAVE GET RID OFF MAX1 IN THE ARRAY SO WE
            ;JUST HAVE TO FIND THE MAX OF THESE VALUES
        MOV BX,OFFSET NUMBERS
        MOV CH,00H
        MOV CL,INDEXN             ;CL=COUNTER OF NUMBERS ARRAY
        MOV MAX2,30H             ;INITIALIZE MAX2 IN ZERO VALUE
NEXTF:
        MOV DL,[BX]
        CMP DL,MAX2
        JLE CONTF          ;IF WE DID FIND A DIGIT LESS
                          ;OR EQUAL THAN MAX2 THEN CHECK THE
                          ;NEXT DIGIT
        MOV MAX2,DL             ;ELSE MAX2 <-THIS DIGIT
CONTF:  INC BX                       ;SHOW TO THE NEXT CELL
        LOOP NEXTF              ;DO THAT FOR THE WHOLE ARRAY
```

```
        PRINT_MESSAGE  MSG3
        PRINT MAX1
        PRINT ' '
        PRINT MAX2

FOUND_MAX: PRINT_STR NEWLINE

EXIT:   JMP START     ;CONTINUOUS FUNCTIONALITY

TERMINATE:
        PRINT_STR BYE
        MOV AX,4C00H  ;RETURN TO DOS
        INT 21H
        MAIN ENDP

CODE ENDS
END MAIN
```