Tutorial – 2 (ALP 8086)
Microprocessor (BCT/BEX/BEL II / II)

TITLE (1) THE ADDITION OF A 100 NATUAL EVEN NUMBERS [SUM = n (n+1)]

```
.MODEL SMALL
.STACK
.DATA
        TEN DW 10
.CODE
MAIN PROC FAR
        MOV AX,@DATA
        MOV DS, AX

        MOV CX, 63H
        MOV AX, 02H
        MOV DX, 04H
L1:     ADD AX, DX
        ADD DX, 02H
        LOOP L1
L2:     MOV DX, 0000
        DIV TEN
        INC CX
        ADD DX, 30H
        PUSH DX
        CMP AX, 00
        JE L3
        JMP L2
L3:     POP DX
        MOV AH, 02H
        INT 21H
        LOOP L3
        MOV AX, 4C00H
        INT 21H
MAIN ENDP
END MAIN
```

TITLE (2) TO CONVERT THE LOWER CASE VOWELS TO UPPER CASE

```
.MODEL SMALL
.STACK 64
.DATA
        MAXLEN DB 100
        ACTCHAR Db ?
        STR DB 100 DUP('$')
        VL DB "aeiou",'$'
        STR1 DB "CONVERTED STRING IS:  ", 0DH, 0AH, '$'
        STR2 DB "ENTER THE STRING: ", 0DH, 0AH, '$'
.CODE
        MAIN PROC FAR
        MOV AX,@DATA
        MOV DS, AX

        LEA DX, STR2
        MOV AH, 09H
        INT 21H

        LEA DX, MAXLEN
        MOV AH, 0AH
        INT 21H

        MOV AH, 02
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H
        MOV CH, 00H
        MOV CL, ACTCHAR
        MOV DI, 00
L2:     MOV SI, 0006H
L1:     MOV AL, STR [DI]
        CMP AL, VL [SI]
        JNZ L3
        CMP AL, 'a'
        JB L3
        CMP AL, 'z'
        JA L3
        SUB AL, 20H
        MOV STR [DI], AL
L3:     SUB SI, 01
        JNZ L1
        INC DI
        LOOP L2

        LEA DX, STR1
```

```
            MOV AH, 09H
            INT 21H

            LEA DX, STR
            MOV AH, 09H
            INT 21H

            MOV AX, 4C00H
            INT 21H
MAIN ENDP
END MAIN
```

TITLE (3) TO COUNT THE NUMBER OF VOWELS IN THE ENTERED STRING.

```
.MODEL SMALL
.STACK 64
.DATA
            MAXLEN Db 100
            ACTCHAR Db ?
            STR DB 100 DUP('$')
            VL DB "AEIOUaeiou",'$'
            STR1 DB "NO. OF VOWELS IS ",'$'
.CODE
            MAIN PROC FAR
            MOV AX,@DATA
            MOV DS, AX
            MOV CX, 00
            MOV DI, 00
            MOV AX, 00
            LEA DX, MAXLEN
            MOV AH, 0AH
            INT 21H
            MOV CH, 00H
            MOV CL, ACTCHAR
            MOV DX, 00
L2:         MOV SI, 000BH
L1:         MOV AL, VL [SI]
            CMP AL, STR [DI]
            JNZ L3

            INC DH
            CMP DH, 0AH
            JB L3
            MOV DH, 00
            INC DL
```

```
L3:     SUB SI, 01
        JNZ L1
        INC DI
        LOOP L2

        MOV BX, DX
        MOV AH, 02H
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H
        LEA DX, STR1
        MOV AH, 09H
        INT 21H

        MOV DX, bX
        ADD DL, 30H
        MOV AH, 02H
        INT 21H

        ADD DH, 30H
        MOV DL, DH
        MOV AH, 02H
        INT 21H
        MOV AX, 4C00H
        INT 21H
MAIN ENDP
END MAIN
```

TITLE (4) TO DISPLAY TEXT AT MIDDLE OF THE SCREEN

```
.MODEL SMALL
.STACK 32
.DATA
        PARALIST LABEL BYTE
        MAXLEN DB 20
        ACTLEN DB ?
        KBNAME DB 21 DUP (' ')
        PROMT DB "NAME?",'$'
.CODE
MAIN PROC FAR
        MOV AX,@DATA
        MOV DS, AX
        MOV ES, AX
        CALL SCREEN
```

```
A1:     MOV DX, 00
        CALL CURSOR
        CALL INPUT
        CALL SCREEN
        CMP ACTLEN, 00
        JE A2
        CALL CENTER
        CALL DISPLAY
        JMP A1
A2:     MOV AX, 4C00H
        INT 21H
MAIN ENDP

SCREEN PROC NEAR
        PUSH AX
        PUSH BX
        PUSH CX
        PUSH DX
        MOV AX, 0600H
        MOV BH, 0C7H
        MOV CX, 00
        MOV DX, 184FH
        INT 10H
        POP DX
        POP CX
        POP BX
        POP AX
        RET
SCREEN ENDP

CURSOR PROC NEAR
        PUSH AX
        PUSH BX
        PUSH DX
        MOV AH, 02H
        MOV BH, 00
        INT 10H
        POP DX
        POP BX
        POP AX
        RET
CURSOR ENDP

DISPLAY PROC NEAR
       PUSH AX
       PUSH DX
       MOV AH, 09H
       LEA DX, KBNAME
```

```
          INT 21H
          POP DX
          POP AX
          RET
DISPLAY ENDP

INPUT PROC NEAR
          PUSH AX
          PUSH DX
          MOV AH, 09H
          LEA DX, PROMT
          INT 21H
          MOV AH, 0AH
          LEA DX, MAXLEN; PARALIST
          INT 21H
          POP DX
          POP AX
          RET
INPUT ENDP

CENTER PROC NEAR
         MOV BL, ACTLEN
         MOV BH, 00
         MOV KBNAME [BX], 0DH
         MOV KBNAME [BX+1], 0AH
         MOV KBNAME [BX+2],'$'
         MOV DL, ACTLEN
         SHR DL, 1
         NEG DL
         ADD DL, 40
         MOV DH, 12
         CALL CURSOR
         RET
CENTER ENDP

END MAIN
```

TITLE (5) TO DISPLAY WORDS OF THE SENTENCE IN SEPARATE LINES

```
DISPLAY MACRO MMM
           MOV AH, 2
           MOV DL, MMM
           INT 21H
ENDM
DISPLAYS MACRO MM
           MOV AH, 9
           LEA DX, MM
```

```
        INT 21H
ENDM
READS MACRO DDD
        MOV AH, 0AH
        LEA DX, DDD
        INT 21H
ENDM
INIT MACRO
        MOV AX,@DATA
        MOV DS, AX
        MOV ES, AX
        ENDM
NEXTLINE MACRO
        MOV AH, 02
        MOV DL, 0AH
        INT 21H
        MOV AH, 02
        MOV DL, 0DH
        INT 21H
 ENDM
FINISH MACRO
        MOV AX, 4C00H
        INT 21H
 ENDM
 ; ----------------------------------------------------------------------

.MODEL
.STACK 64
.DATA
        SR DB "ENTER THE STRING: ",'$'
        MAXL DB 0feh
        ACTL DB ?
        STR DB 0ffh DUP(?)
.CODE
MAIN PROC FAR
        INIT
        NEXTLINE
        DISPLAYS SR
        NEXTLINE
        READS MAXL
        NEXTLINE
        NEXTLINE

        MOV BX, 00
        MOV CL, ACTL
        MOV CH, 00
L1:     CMP STR [BX], 32
        JNZ S1
        CMP STR [BX+1], 32
        JNZ S2
S3:     INC BX
        CMP CL, 01
        JZ LL2
        SUB CL, 01
LL2:    CMP STR [BX], 32
        JZ S3
        DEC BX
S2:     NEXTLINE
        INC BX
S1:     DISPLAY STR [BX]
        INC BX
        LOOP L1
        NEXTLINE
        FINISH
MAIN ENDP
END MAIN
```

TITLE (6) TO COUNT THE NUMBER OF WORDS PRESENTED IN THE ENTERED SENTENCE.

```
.MODEL SMALL
.STACK 64
.DATA
        MAXLEN Db 100
        ACTCHAR Db ?
        STR DB 101 DUP('$')
        STR1 DB "NO. OF WORDS IS ",'$'
.CODE
MAIN PROC FAR
        MOV AX, @DATA
        MOV DS, AX
        MOV CX, 00
        MOV BX, 00
        MOV AX, 00
        LEA DX, MAXLEN
        MOV AH, 0AH
        INT 21H


        MOV CH, 00H
        MOV Cl, ACTCHAR
        MOV DX, 0100H
        CMP STR [0], ' '
        JNZ L1
        SUB DH, 01
```

```
L1:     CMP STR [BX], ' '
        JNZ L3
L2:     INC BX
        DEC CX
        CMP STR [BX], ' '
        JZ l2
        INC DH
        CMP DH, 0AH
        JB L3
        MOV DH, 00
        INC DL
L3:     INC BX
        LOOP L1
        CMP STR [BX-1], ' '
        JNZ L4
        SUB DH, 01
        JNC L4
        SUB DL, 01
        ADD DH, 0AH

  L4:   MOV BX, DX
        MOV AH, 02H
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H
        LEA DX, STR1
        MOV AH, 09H
        INT 21H

        MOV DX, BX
        ADD DL, 30H
        MOV AH, 02H
        INT 21H
        ADD DH, 30H
        MOV DL, DH
        MOV AH, 02H
        INT 21H
        MOV AX, 4C00H
        INT 21H
  MAIN ENDP
  END MAIN
```

TITLE (7) TO GET NUMBER FROM THE USER AND SUM THE ODD NUMBERS FROM 1 TO NTH TERM AND DISPLAY SUM. [SUM = n^2]

```
.MODEL SMALL
.STACK
.DATA
        TEN DW 0010
        TIN DW 0000
        SUM DW 0000
.CODE
MAIN PROC FAR
        MOV AX,@DATA
        MOV DS,     AX
        MOV DX,     0000
L4:     MOV AH,     01H
        INT 21H
        CMP AL, 0DH   ; COMPARE WITH ENTER
        JE L5
        SUB AL, 30H
        MOV AH, 00
        MOV TIN, AX
        MOV AX, SUM
        MUL TEN
        ADD AX, TIN
        MOV SUM, AX
        JMP L4
L5:     MOV CX, SUM
        MOV AX, 00H
        MOV DX, 01H
L1:     ADD AX, DX
        ADD DX, 02H
        LOOP L1
L2:     MOV DX, 00
        DIV TEN
        INC CX
        PUSH DX
        CMP AX, 00
        JNE L2
        MOV AH, 02H
        MOV DL, 0DH
        INT 21H
        MOV DL, 0AH
        INT 21H
```

```
L3:     POP DX
        ADD DX, 30H    ; ADD DX, 48
        INT 21H
        LOOP L3
        MOV AX, 4C00H
        INT 21H
MAIN ENDP
END MAIN
```

TITLE (8) ADDITION OF NUMBERS FROM 1 TO N, WHERE N BEING ENETRED NUMBER BY USER HAVING 3 DIGITS.

```
DISPLAY MACRO MMM
        MOV AX, MMM
        MOV CX, 00
        MOV BX, 0AH
SS1:    MOV DX, 00
        DIV BX
        PUSH DX
        INC CX
        CMP AX, 00
        JNZ SS1
SS2:    POP DX
        ADD DL, 30H
        MOV AH, 02H
        INT 21H
        LOOP SS2
ENDM

DISPLAYS MACRO MM
        MOV AH, 09H
        LEA DX, MM
        INT 21H
ENDM

READ MACRO
        MOV CX, 03
LL1:    MOV AH, 01
        INT 21H
        CMP AL, 0DH
        JE LP
        SUB AL, 30H
        PUSH AX
        MOV AL, 10
        MUL NUM
        MOV NUM, AL
        POP AX
        ADD NUM, AL
        INC COUNT
LP:     LOOPNE LL1
ENDM


INIT MACRO
        MOV AX,@DATA
        MOV DS, AX
        MOV ES, AX
ENDM

NEXTLINE  MACRO
        MOV AH, 02
        MOV DL, 0AH
        INT 21H
        MOV AH, 02
        MOV DL, 0DH
        INT 21H
ENDM

FINISH MACRO
        MOV AX, 4C00H
        INT 21H
ENDM
 ; ----------------------------------------------------------------------


.MODEL SMALL
.STACK 64
.DATA
        STR DB "ENTER THE THREE NUMBERS: (LESS THAN 256 ) : ",'$'
        STR1 DB "THE SUM UPTO THE NUMBER IS: $"
        NUM DB 0
        COUNT DB 0
        SUM DW ?
.CODE
START PROC FAR
        INIT
        DISPLAYS STR
        READ
        NEXTLINE
        MOV CH, 00
        MOV CL, NUM
        MOV AX, 00
```

```
S1:     ADD AX, CX
        LOOP S1
        MOV SUM, AX
        DISPLAYS STR1
        DISPLAY SUM

        FINISH
START ENDP
END START
```

TITLE (9) DISPLAY THE STRING AT THE MIDDLE OF THE SCREEN ONE LETTER AT ONE LINE WITH BLUE FOREGROUNG AND WHITE BACKGROUND.

```
.MODEL SMALL
.STACK 32
.DATA
        MAXLEN DB 23
        ACTLEN DB ?
        STR1 DB 24 DUP('$')
.CODE
MAIN PROC FAR
        MOV AX,@DATA
        MOV DS, AX
        MOV ES, AX

        MOV AH, 0AH
        LEA DX, MAXLEN
        INT 21H

        MOV AX, 0600H
        MOV BH, 71H
        MOV CX, 00
        MOV DX, 184FH
        INT 10H

        MOV BX, 00
        MOV CH, 00
        MOV CL, ACTLEN

        MOV AL, 24
        SUB AL, ACTLEN
        SHR AL, 01
        MOV DH, AL
```

```
L1:     MOV AH, 02H
        MOV BH, 00
        MOV DL, 40
        INT 10H

        MOV DL, STR1 [BX]
        MOV AH, 02
        INT 21H

        INC DH
        INC BX
        LOOP L1

        MOV AX, 4C00H
        INT 21H
MAIN ENDP
END MAIN
```

TITLE (10) SORTING ARRAYS OF NUMBERS AND DISPLAY EACH NUMBER IN SEPARATE LINE

```
SWAP MACRO N1, N2
        MOV AL, N1
        MOV AH, N2
        MOV N1, AH
        MOV N2, AL
        MOV BX, 01
ENDM

NEXTLINE MACRO
        MOV AH, 02
        MOV DL, 0AH
        INT 21H
        MOV DL, 0DH
        INT 21H
ENDM

DISPLAY MACRO NUMB
        MOV Al,NUMB
        MOV AH, 00
        MOV BL, 10H
        DIV BL
        MOV DL, AL
        MOV DH, AH
        CMP DL, 0AH
        JB L1
        ADD DL, 07
```

```
L1:     ADD DL, 30H
        MOV AH, 02
        INT 21H
        MOV DL, DH
        CMP DL, 0AH
        JB L2
        ADD DL, 07
L2:     ADD DL, 30H
        MOV AH, 02
        INT 21H
ENDM
;-----------------------------------------------------------------------
.MODEL SMALL
.STACK 64
.DATA
        NUM DB 99H,98H,88H,77H,66H,55H,44H,33H,22H,11H
        STR DB "AFTER SORTINGS:$"
.CODE
MAIN PROC FAR
        MOV AX,@DATA
        MOV DS, AX
S3:     MOV CX, 10
        MOV BX, 00
        MOV SI, 00
S2:     MOV AH, NUM [SI]
        CMP AH, NUM [SI+1]
        JB S1
        JE S1
        SWAP NUM [SI], NUM [SI+1]
S1:     INC SI
        LOOP S2
        CMP BX, 00
        JNZ S3
        LEA DX, STR
        MOV AH, 09
        INT 21H
        NEXTLINE
        MOV CX, 10
        MOV SI, 00
SSS:    DISPLAY NUM [SI]
        NEXTLINE
        INC SI
        LOOP SSS

        MOV AX, 4C00H
        INT 21H
MAIN ENDP
END MAIN
```