

CSE331L_4 – Loops & Statements

1. Write an ASM code to print upper case letter from A to Z

```

.MODEL SMALL
.STACK 100H

.DATA
    PROMPT    DB    "The Upper-Case Letters from A to Z are : $"

.CODE
MAIN PROC
    MOV AX, @DATA    ; initialize DS
    MOV DS, AX

    LEA DX, PROMPT    ; load and print PROMPT
    MOV AH, 9
    INT 21H

    MOV CX, 26        ; initialize CX

    MOV AH, 2         ; set output function
    MOV DL, 65        ; set DL with A

@LOOP:                ; loop start
    INT 21H           ; print character
    INC DL            ; increment DL to next ASCII character
    DEC CX            ; decrement CX

    JNZ @LOOP         ; jump to label @LOOP if CX is 0
    MOV AH, 4CH        ; return control to DOS
    INT 21H

    MAIN ENDP
END MAIN

```

2. Write an ASM code to read a letter in Upper case and print it after converting it to lower case.

```

I.MODEL SMALL
.STACK 100H

.DATA
    PROMPT_1 DB "Enter the Upper-Case Letter : $"
    PROMPT_2 DB 0DH,0AH, "The Lower-Case Letter is : $"

.CODE
MAIN PROC
    MOV AX, @DATA          ; initialize DS
    MOV DS, AX

    LEA DX, PROMPT_1       ; load and print PROMPT_1
    MOV AH, 9
    INT 21H

    MOV AH, 1              ; read a letter
    INT 21H

    MOV BL, AL             ; save the letter in BL

    LEA DX, PROMPT_2       ; load and print PROMPT_2
    MOV AH, 9
    INT 21H

    OR BL, 20H             ; convert an upper-case letter to lower
                           ; case letter

    MOV AH, 2              ; print the Lower-case letter
    MOV DL, BL
    INT 21H

    MOV AH, 4CH            ; return control to DOS
    INT 21H
MAIN ENDP
END MAIN

```

3. Write an ASM code to print 2's compliment of a given number

```
Data Segment
    num db 00000010B
Data Ends

Code Segment
    Assume cs:code, ds:data

Begin:
    mov ax, data
    mov ds, ax
    mov es, ax
    mov ah, 0000h
    mov al, num
    NOT al
    mov bl, al
    adc al, 00000001B
    mov bl, al

Exit:
    mov ax, 4c00h
    int 21h
    |
Code Ends
End Begin
```

4. Write an ASM code to test if a number is even or odd (from 0-9)

```

.MODEL SMALL
.STACK 100H

.DATA
    PROMPT_1 DB "Enter the number from 0 to 9 : $"
    PROMPT_2 DB 0DH,0AH,"The number is : $"

.CODE
MAIN PROC
    MOV AX, @DATA        ; initialize DS
    MOV DS, AX

    LEA DX, PROMPT_1     ; load and print PROMPT_1
    MOV AH, 9
    INT 21H

    MOV AH, 1             ; read a digit
    INT 21H

    MOV BL, AL            ; save the digit in BL

    LEA DX, PROMPT_2     ; load and print PROMPT_2
    MOV AH, 9
    INT 21H

    TEST BL, 01H         ; check the digit for even or odd
    JNE @ODD              ; jump to label @ODD if the number
                          ; is odd

    MOV AH, 2             ; print the letter 'E'
    MOV DL, "E"
    INT 21H

    TEST BL, 01H         ; check the digit for even or odd
    JNE @ODD              ; jump to label @ODD if the number
                          ; is odd

    MOV AH, 2             ; print the letter 'E'
    MOV DL, "E"
    INT 21H

    JMP @EXIT             ; jump to the label @EXIT

@ODD:                    ; jump label
    MOV AH, 2             ; print the letter '0'
    MOV DL, "0"
    INT 21H

@EXIT:                   ; jump label
    MOV AH, 4CH           ; return control to DOS
    INT 21H

MAIN ENDP
END MAIN

```