**i)Byte and word data transfer in different addressing modes**

DATA SEGMENT

DATA1 DB 23H

DATA2 DW 1234H

DATA3 DB 0H

DATA4 DW 0H

DATA5 DW 2345H,6789H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA ;Initialize DS to point to start of the memory

MOV DS,AX ;set aside for storing of data

MOV AL,25H ;copy 25H into 8 bit AL register

MOV AX,2345H ;copy 2345H into 16 bit AX register

MOV BX,AX ;copy the content of AX into BX register(16 bit)

MOV CL,AL ;copy the content of AL into CL register

MOV AL,DATA1 ;copies the byte contents of data segment memory

;location DATA1 into 8 bit AL

MOV AX,DATA2 ;copies the word contents of data segment memory

;location DATA2 into 16 bit AX

MOV DATA3,AL ;copies the AL content into the byte contents of data

;segment memory location DATA3

MOV DATA4,AX ;copies the AX content into the word contents of

;data segment memory location DATA4

MOV BX,OFFSET DATA5 ;The 16 bit offset address of DS memeory location

; DATA5 is copied into BX

MOV AX,[BX] ; copies the word content of data segment

;memory location addressed by BX into

;AX(register indirect addressing)

MOV DI,02H ;address element

MOV AX,[BX+DI} ; copies the word content of data segment

;memory location addressed by BX+DI into

;AX(base plus indirect addressing)

MOV AX,[BX+0002H] ; copies the word content of data segment

;(16 bit)

MOV AL,[DI+2] ;register relative addressing

MOV AX,[BX+DI+0002H] ;copies the word content of data segment

;memory location addressed by BX+DI+0002H

;into AX(16 bit)

MOV AH,4CH ; Exit to DOS with function call 4CH

INT 21H

CODE ENDS ; Assembler stop reading

END START

**ii) Hello world**

org 100h

jmp start

msg: db "Hello, World!", 0Dh,0Ah, 24h

start: mov dx, msg

mov ah, 09h

int 21h

mov ah, 0

int 16h

ret

**iii) Block Interchange**

DATA SEGMENT

X DB 01H,02H,03H,04H,05H

Y DB 11H,12H,13H,14H,15H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:MOV AX,DATA

MOV DS,AX

MOV CX,05H ; Load the counter

LEA SI,X ; SI pointed to the source location x

LEA DI,Y ; DI pointed to the destination location y

UP: MOV BL,[SI] ; Move the SI content to BL register

MOV AL,[DI] ; Move the DI content to AL register

MOV [SI],AL ; Move AL register content to content of SI

MOV [DI],BL ; Move BL register content to content of DI

INC SI ; Update SI and DI

INC DI

DEC CX ; Decrement the counter till it becomes zero

JNZ UP

MOV AH,4CH

INT 21H

CODE ENDS

END START

**iv ) 16 Bit Addition**

DATA SEGMENT

NUM DW 1234H, 0F234H

SUM DW 2 DUP(0)

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS:DATA

START: MOV AX,DATA

MOV DS,AX

MOV AX,NUM ; First number loaded into AX

MOV BX,0H ; For carry BX register is cleared

ADD AX,NUM+2 ; Second number added with AX

JNC DOWN ; Check for carry

INC BX ; If carry generated increment the BX

DOWN: MOV SUM,AX ; Storing the sum value

MOV SUM+2,BX ; Storing the carry value

MOV AH,4CH

INT 21H

CODE ENDS

END START