**M.I.T. LAB Assignment – 11**

**U19CS012**

1. Write a Program for data transfer using different addressing modes.

TASM Code:

.model small

.stack 100

.8086

.code

mov AX,@data

mov DS,AX

; 1) IMMEDIATE ADDRESSING MODE

mov CL, 12H     ; Moves 12 immediately into CL register

mov BX, 1234H   ; Moves 1234 immediately into BX register

; 2) REGISTER ADDRESSING MODE

mov CL, DL      ; Moves data of DL register into CL register

mov AX, BX      ; Moves data of BX register into AX register

; 3) DIRECT ADDRESSING MODE

mov CL, [4321H] ; Moves data from location 4321H in the data segment into CL

                ; Phy Addr = DS \* 10H + 4321

mov CX, [4320H] ; Moves data from location 4320H and 4321H

                ; in the data segment into CL and CH resp.

; 4) INDIRECT ADDRESSING MODES

; 4.1) REGISTER INDIRECT ADDRESSING MODE

MOV CL, [BX]    ; Moves a byte from the address pointed by BX in Data

                ; Segment into CL.

                ; Physical Address calculated as DS \* 10H + BX

; 4.2) REGISTER RELATIVE ADDRESSING MODE

MOV CL, [BX+4]  ; Moves a byte from the address pointed by BX+4 in

                ; Data Seg to CL.

                ; Physical Address: DS \* 10H + BX + 4H

; 4.3) BASE INDEXED ADDRESSING MODE

MOV CL, [BX+SI] ; Moves a byte from the address pointed by BX+SI in Data Segment to CL.

                ; Physical Address: DS \* 10H + BX + SI

; 4.4) BASE RELATIVE PLUS INDEX ADDRESSING MODE

MOV CL, [BX+DI+20]; Moves a byte from the address pointed by BX+SI+20H in Data Segment to CL.

                  ; Physical Address: DS \* 10H + BX + SI+ 20H

; 5) IMPLIED ADDRESSING MODE

STC ; Sets the Carry Flag

CLD ; Clears the Direction Flag

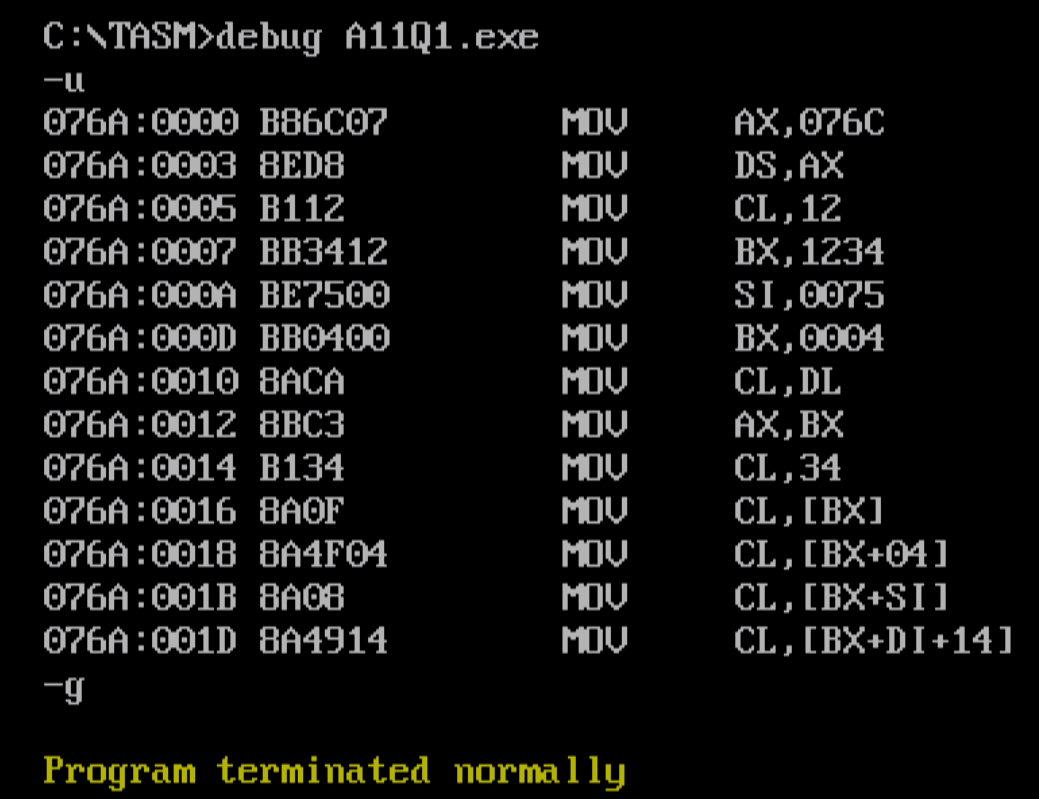
; HLT

mov AX,4c00h

int 21H

end

Output:



2. Write Program to move data from source to destination using indirect addressing mode (Block Move without overlap).

TASM Code:

.model small

.stack 100

.8086

.data

; Number of Elements in Array

n dw 6

; The Contents of Array

arr dw 11H, 22H, 33H, 44H, 55H, 66H

; The ans Array

ans dw ?

.code

mov ax,@data

mov ds,ax

; Initialize the Counter

mov cx, n

; Initialize the Stack Pointer

mov si, 0000H

; Intialize the Destination Pointer

mov di, 0000H

transfer:   mov ax, arr[si]

          inc si

          inc si

          mov ans[di], ax

          inc di

          inc di

          loop transfer ; DCR cx & if cx!=0 goto trnafer

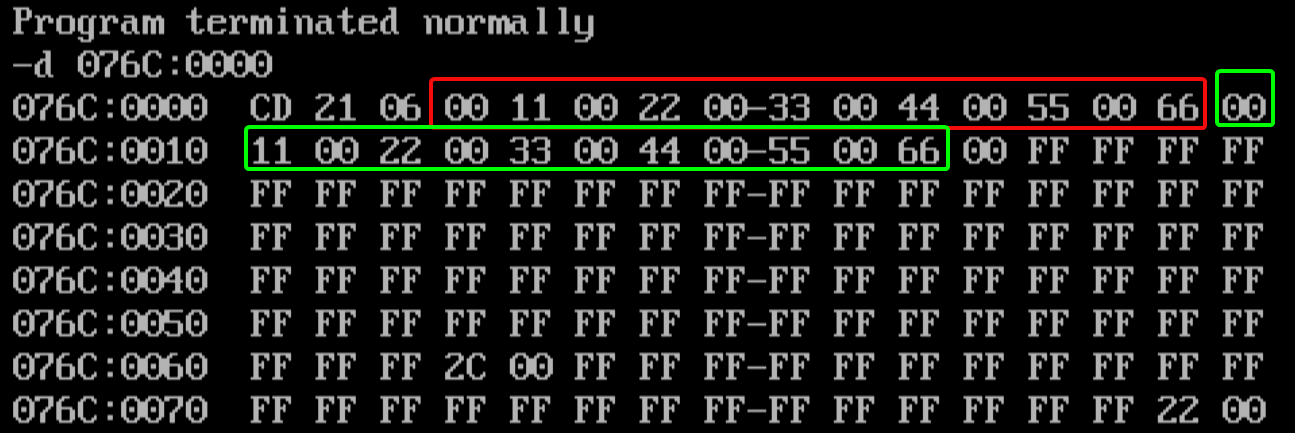
; HLT

mov ax,4C00H

int 21h

end

Output:



3. Write a Program to move a block of data from source to destination (With overlap in either direction).

TASM Code:

.model small

.stack 100

.8086

.data

; 10 Data Bytes

x db 01h,02h,03h,04h,05h,06h,07h,08h,09h,0ah

y db 10 dup(?)

.code

mov ax,@data

mov ds,ax

mov ES,ax

mov si,offset x

mov di,offset y

mov cx, 000ah

add si, 0009h

add di, 0004h

up:     mov al, [si]

        mov [di], al

        dec si

        dec di

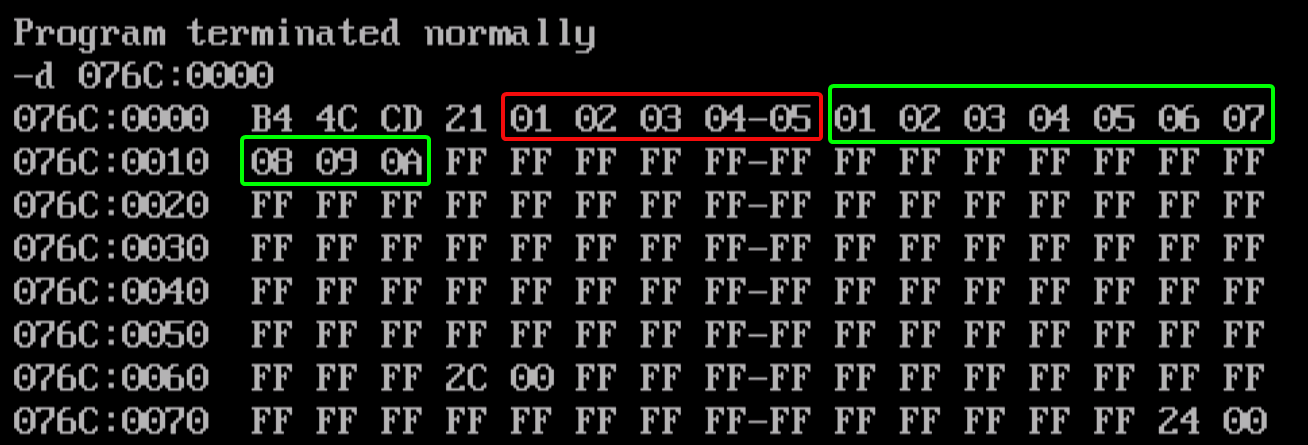
        loop up

mov ax,4c00h

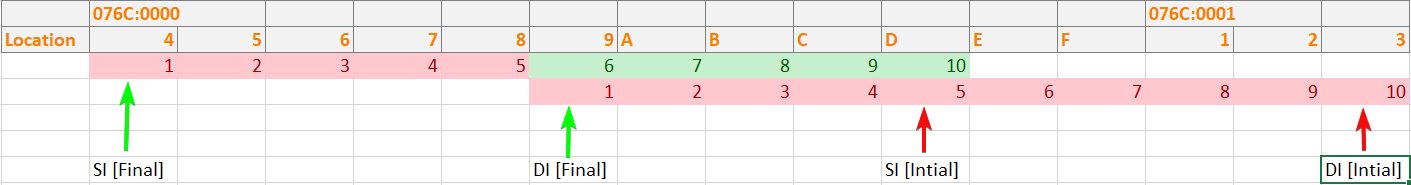
int 21h

end

Output:



Visual Understanding of Above Process



4. Write a Program to interchange two blocks of data.

TASM Code:

.model small

.stack 100

.8086

.data

; Number of Bytes in Each Block

bytecnt equ 05

; Source Block Data

src db 11H, 22H, 33H, 44H, 55H

; Destination Block Data

dst db 66H, 77H, 88H, 99H, 0AAH

.code

mov ax, @data

mov ds, ax

; Intialize SI, DI and Counter

lea si, src

lea di, dst

mov cl, bytecnt

up: mov al,[si]     ; To Swap Store Value in Temp Variable

    mov bl,[di]

    mov [si],bl     ; Store the Old Destination Value in [SI]

    mov [di],al     ; Store the Old Source Value in [DI]

    inc si

    inc di

    dec cl

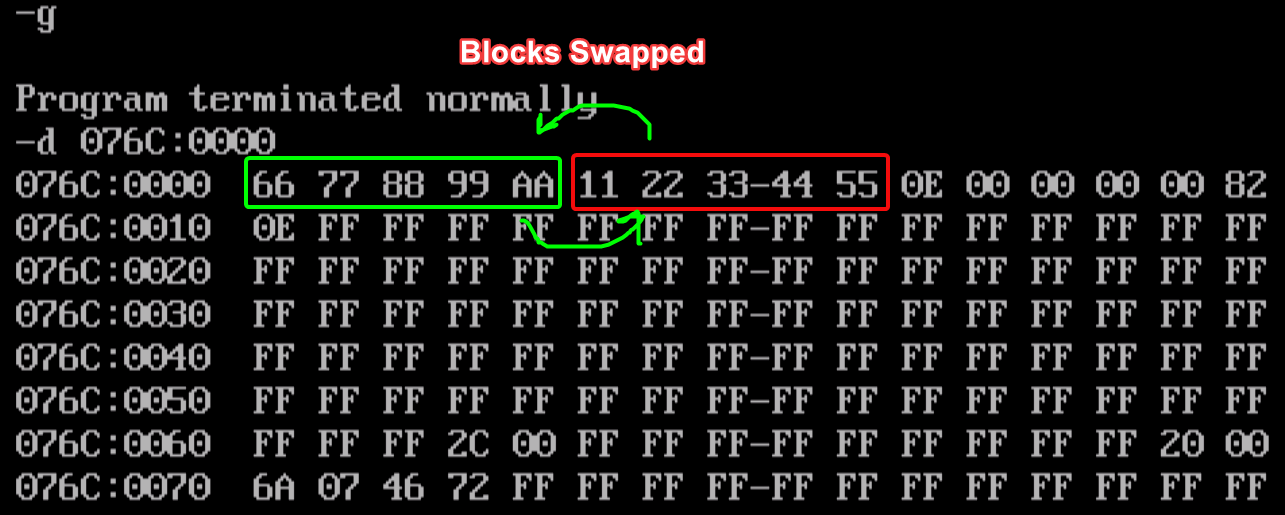
    jnz up

mov ax,4C00H

int 21h

end

Output:



SUBMITTED BY:

BHAGYA VINOD RANA

[***U19CS012***]