**Question 1 : Write a assembly program to convert Binary to ASCII**

.model small ;model small tells the assembler that you intend to use the small memory model - one

;code segment, one data segment and one stack segment

.data ;Start of the data segment

INPUT DB 10,13 , 'ENTER BINARY NO: $' ; $ -> end of string, 10 -> Line Feed, ;13 -> Carriage Return

OUTPUT DB 10,13, 'THE ASCII CHARACTER IS:$'

ARR DB ? ;Define a variable ARR with an uncertain default value

.code ;Defines start of Code Segment

.startup ;Indicates startup code. Startup Code is a small block of assembly language code that

;prepares the way for the execution of software written in a high-level language

MOV AH,09 ;Display a character string

MOV DX,OFFSET INPUT ;Puts the offset of ?INPUT' to DX register. Why DX? DX can address memory data

;and thus has the memory address of the character string

INT 21H ;DOS Interrupt

MOV BL, 00H ;Clears BL to 00H

MOV CL,08H ;Set CL to 08H INPUT1: MOV AH,01H ;reads the input from keyboard

INT 21H ;DOS interrupt

SUB AL,30H ;The ASCII values of numbers and actual number has difference of 30.

SHL BL,1 ;Performs a logical left shift on BL

ADD BL,AL ;Adds the content of AL to BL

LOOP INPUT1 ;Loops INPUT1

MOV AH,09H ; Display a character string

LEA DX,OUTPUT ;Load the effective address, offset of OUTPUT in DX

INT 21H ;DOS Interrupt

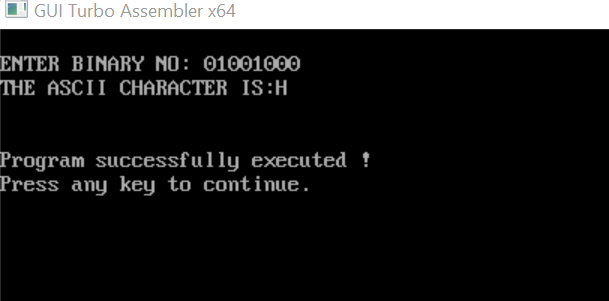
MOV AH,02H ;display a single character

MOV DL,BL ;Copies the content of BL to DL

INT 21H ;DOS Interrupt

.EXIT ;exits to DOS

END ;ends the program



**Question 2 : Write a assembly program for searching**

.model small ;model small tells the assembler that you intend to use the small memory model - one

;code segment, one data segment and one stack segment

.386 ;microprocessor 80386 instructions will be used

.data ;Start of the data segment

ARRAY DW 20 DUP (?) ;Define an array of each element word. Duplicate (?) 20 times. This creates an array

;[?|?|?|...]

DATA1 dw 0000H ;Define word

success db 10,13,"Element is present in the array $"

fail db 10,13,"Element is not present in the arary $"

msg db 10,13,"Enter the size of the array :: $"

msg2 db 10,13,"Enter the array :: $"

msg3 db 10,13,"Enter the element to be searched :: $"

.code ;Defines start of Code Segment

.startup ;Indicates startup code. Startup Code is a small block of assembly language code that

;prepares the way for the execution of software written in a high-level language

MOV AH,09 ;Display a character string

MOV DX,OFFSET msg ;Puts the offset of 'msg' to DX register. Why DX? DX can address memory data and

;thus has the memory address of the character string

INT 21H ;DOS interrupt

MOV AH,01 ;Input a character

INT 21H ;DOS Interrupt

SUB AL,30H ;Subtract 30H from AL. The ASCII values of numbers and actual number has

;difference of 30.

MOV AH,0 ;Copies 0 to AH to remove garbage value from AH

MOV CX,AX ;Copies the data of AX to CX. We got the count - number of elements we want in

;array

MOV DATA1,AX ;Copy data of AX to DATA1

MOV AH,09 ;Display a character string

MOV DX,OFFSET msg2 ;Puts the offset of 'msg2' to DX register. Why DX? DX can address memory data and

;thus has the memory address of the character string

INT 21H ;DOS interrupt

MOV AH,0 ;Clearing the data of AH register

MOV SI, 0 ;Clearing the data of SI register

MOV BX, OFFSET ARRAY ;Copies the offset of array to BX. BX holds the offset address of a location in the

;memory system.

L1: MOV DL, 0AH ;jump onto next line. 0AH -> Line Feed. '\n'

MOV AH, 02H ;Display a single character

INT 21H ;DOS Interrupt

MOV DX, SI ;input element of the array

MOV AH, 01H ;Read the input from keyboard

INT 21H ;DOS interrupt

SUB AL,30H ;Subtract 30H from AL. The ASCII values of numbers and actual number has

;difference of 30.

MOV SI, DX ;Copies value of DX to SI register

MOV [BX + SI], AX ;Copies value of AX register to [BX+SI]

INC SI ;Increases SI value

LOOP L1 ;Loop label L1

MOV CX,DATA1 ;Copies the value of DATA1 variable into the Count register

MOV AH,09 ;Display a character string

MOV DX,OFFSET msg3 ;Puts the offset of 'msg3' to DX register. Why DX? DX can address memory data and

;thus has the memory address of the character string

INT 21H ;DOS interrupt

MOV AH,01 ;Enter element to be searched

INT 21H ;DOS Interrupt

SUB AL,30H ;Subtract 30H from AL. The ASCII values of numbers and actual number has

;difference of 30.

MOV SI, 0 ;Clearing the data of SI register

MOV BX, OFFSET ARRAY ;Copies the offset of array to BX. BX holds the offset address of a location in the

;memory system.

L2: CMP [BX + SI], AL ;linear search loop

JZ L3 ;jump if element is found

INC SI ;Increments the value of SI

LOOP L2 ;Loops label L2

MOV AH,09 ;Display a character string

MOV DX,OFFSET fail ;Puts the offset of 'fail' to DX register. Why DX? DX can address memory data and

;thus has the memory address of the character string

INT 21H ;DOS interrupt

MOV AH, 4CH ;Exit to DOS

INT 21H ;DOS interrupt

L3: MOV AH, 09H ;Display a character string

MOV DX,OFFSET success ;Puts the offset of 'success' to DX register. Why DX? DX can address memory data

;and thus has the memory address of the character string

INT 21H ;DOS interrupt

MOV AH, 4CH ;Exit to DOS

INT 21H ;DOS interrupt

.EXIT ;Exits to DOS

END ;Ends the program

