St. Francis Institute of Technology

Class: SE-ITA/ITB Semester: IV; A.Y. 2023-2024 Subject: Microprocessor Lab

Experiment – 6: Linear search for a given element in an array

1. Aim:

Write an ALP to search for a given element in an array.

2. Requirements

DOSBox (an x86 emulator with DOS), Turbo Assembler, Turbo Debugger

3. Pre-Experiment Exercise

Algorithm:

- a. Initialize an array of 8 bit numbers and the number that is to be searched.
- b. Use a MACRO to display messages on the output screen.
- c. Define a procedure to compare the number with all elements in the array.
- d. If the number equals any of the elements of the array, then print "Number found" else print "Number not found"

4. Laboratory Exercise:

Procedure:

- a. Open DOSbox and go to TASM.
- b. Open a new document using the command edit <filename>.asm
- c. Write the Program and save the changes to the same file.
- d. Assemble the program using the command tasm <filename.asm>
- e. If any errors are displayed, then change the code in <filename>
- f. If no errors are displayed, execute the command tlink <filename>.obj to create the executable file.
- g. Next execute the command <filename>
- h. Check the result displayed.

5. Post Experiment Exercise:

a. Results/Calculations/Observations:

Attach screenshots, one of number found in the array and another of number not found in the array along with the ALP.

b. Questions:

- i. Write an ALP in TASM to find the largest number in an array. Attach appropriate screenshots.
- ii. Explain different types of loop instructions used in 8086.
- iii. Explain ALP commands to generate interrupt.

c. Conclusion:

Write the conclusion/comments based on the experiment performed and the output obtained.

d. References:

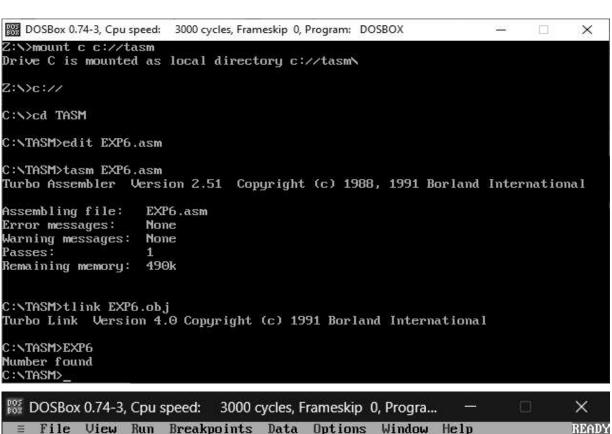
Mention two book references and two web references.

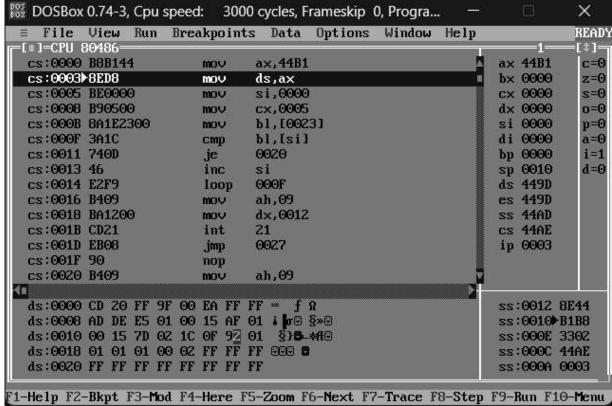
```
Name: Vishal Rajesh Mahajan
Class: SE INFT A
Code:
model small
stack 10h
data segment
        arr dB 25h,13h,06h,0Ah,43h
        msg1 dB "Number found $"
        msg2 dB "Number not found $"
        element dB 06h : 20h
data ends
print macro msg
        mov ah,09h
        lea dx,msg
        int 21h
endm
code segment
        assume cs:code,ds:data
start:
        mov ax, data
        mov ds, ax
        lea si, arr
        mov cx, 05h
        mov bl, element
     up:cmp bl, [si]
        je found
        inc si
        loop up
        print msg2
        jmp exit
  found:print msg1
  exit: mov ah, 4ch
        int 21h
code ends
end start
```

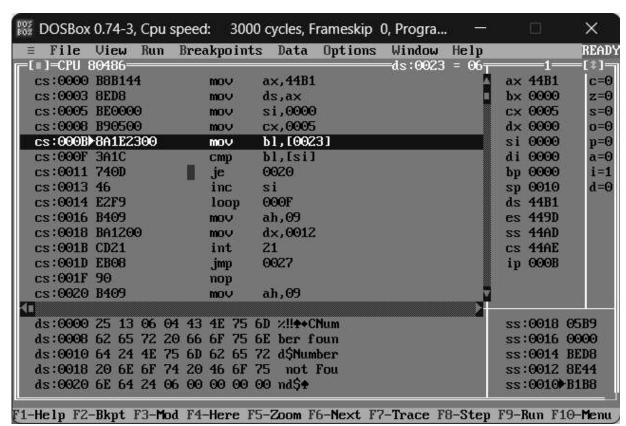
Roll No: 63

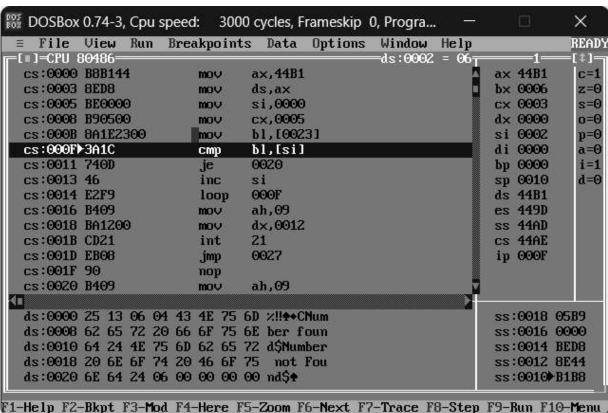
EXP No.: 6

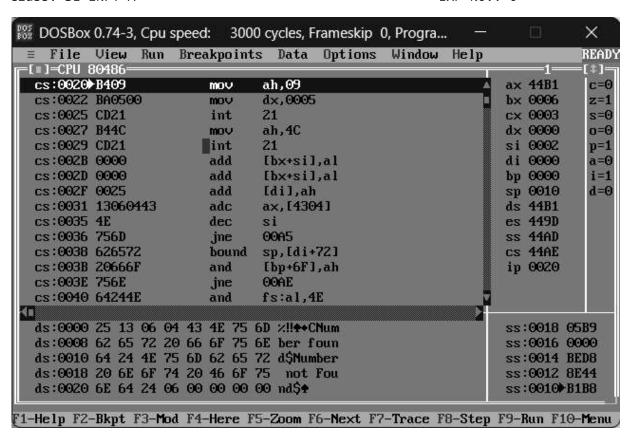
When Number is 06

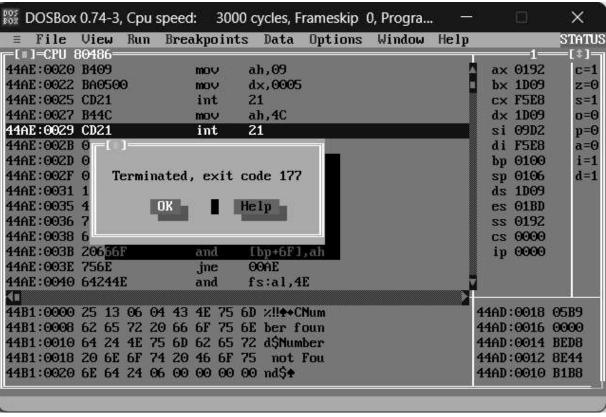




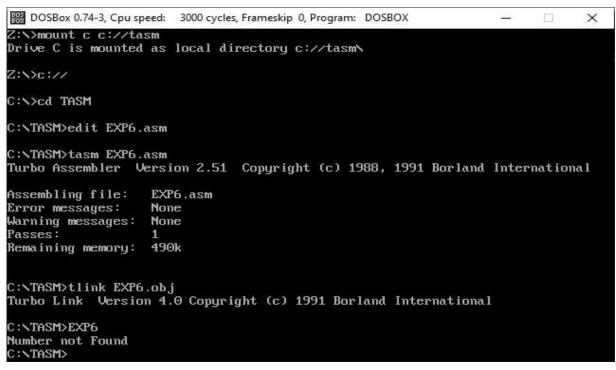


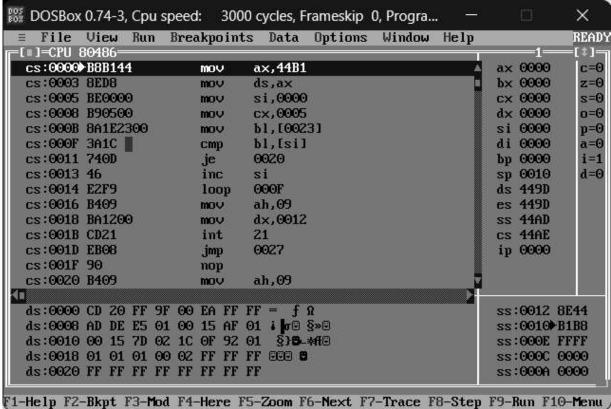


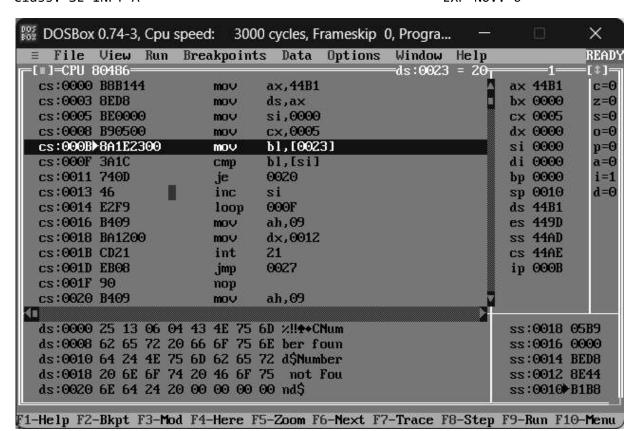


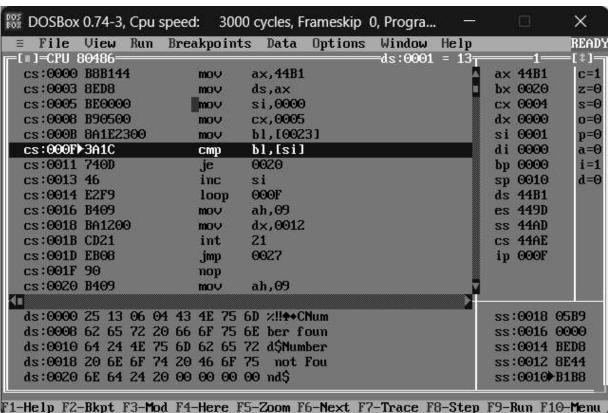


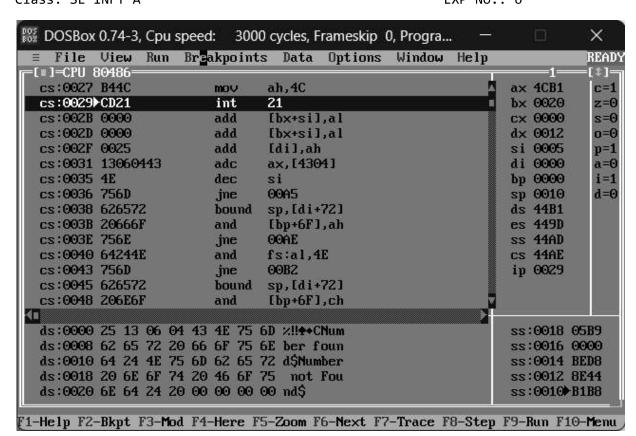
When number is 20

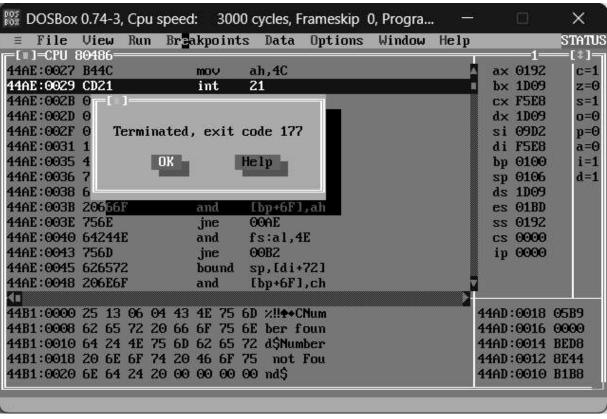








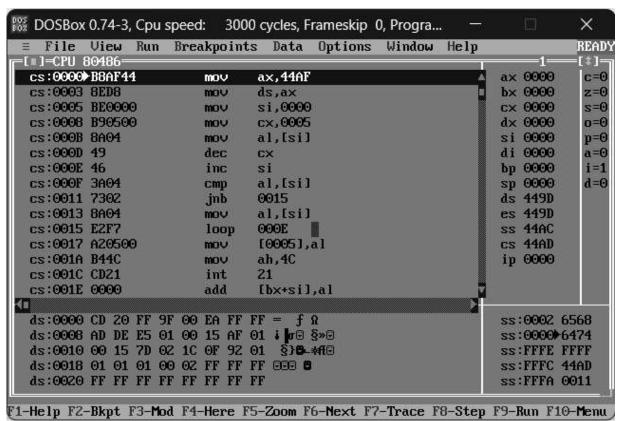




<u>POST EXP: Write an ALP in TASM to find the largest number in an array.</u>
<u>Attach appropriate screenshots.</u>

```
Code:
model small
data segment
     arr db 25h, 13h, 06h, 01h, 43h
     element dB 00h
data ends
code segment
     assume cs:code, ds:data
start:
     mov ax, data
     mov ds, ax
     lea si, arr
     mov cx, 05h
     mov al, [si]
     dec cx
     up: inc si
     cmp al, [si]
     jnc next
     mov al, [si]
  next: loop up
     mov element, al
     mov ah, 4ch
     int 21h
code ends
end start
```

Output:



File Vie		akpo i nt	s Data Options Window	Help	RE
□ 1=CPU 8048 cs:0000 B8A		mov	ax,44AF	ax 44AF	=[‡
cs:0003 8ED	T00202	mov	ds,ax	b× 0000	z
cs:0005 BE0		mov	si,0000	cx 0000	S
cs:0008 B90		mov	cx,0005	d× 0000	o
cs:000B 8A0	potential i	mov	al,[si]	si 0000	p
cs:000D 49	· -	dec	CX	di 0000	a
cs:000E 46		inc	si	bp 0000	i
cs:000F 3A0	4	cmp	al,[si]	sp 0000	d
cs:0011 730		jnb	0015	ds 44AF	
cs:0013 8A0		mov	al,[si]	es 449D	
cs:0015 EZF	7	loop	000E	ss 44AC	
cs:0017 A20	500	mov	[0005],al	cs 44AD	
cs:001A B44		mov	ah,4C	ip 0005	
cs:001C CD2	1	int	21		
cs:001E 000	Θ	add	[bx+si],al		
ds:0000 25	13 06 01 43	00 B4	09 ×!! • ©C 4∘	₫ ss:0008 6	864
			90 \$ =!δ 0 É	ss:0006 7	42E
ds:0010 B4	09 BA 05 00	CD 21	B4 0 0 = !-	ss:0004 7	'06C
ds:0018 4C	CD 21 00 00	00 00	⊙⊙ L=!	ss:0002 6	568
ds:0020 25	13 06 04 43	4E 75	6D ×!! • CNum	ss:0000⊁6	474

