VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Microprocessor and Interfacing

CSE2006

Final Lab Assessment Test

Slot: L3+L4

Name: Kulvir Singh

Register Number: 19BCE2074

Question:

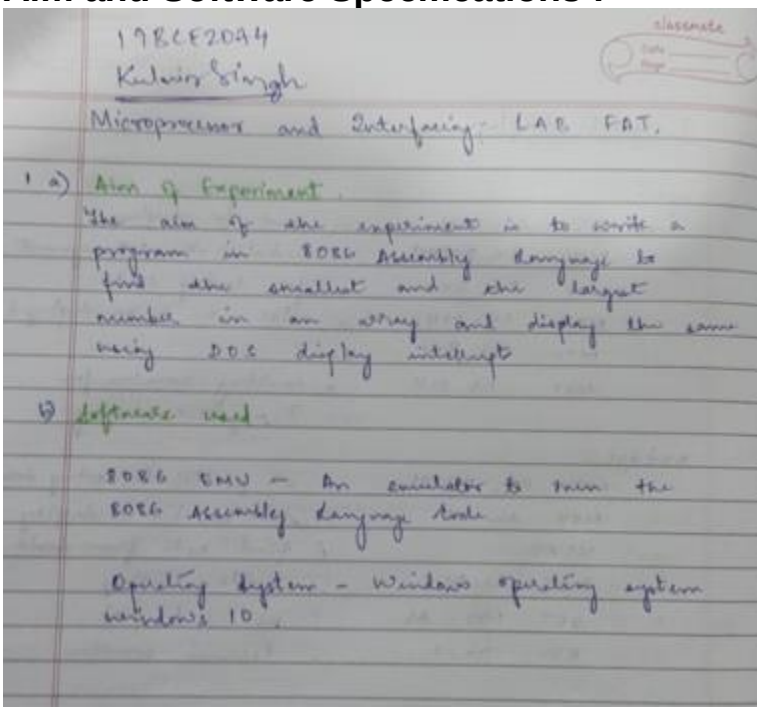
FINAL ASSESSMENT TEST
MICROPROCESSOR & INTERFACING
SCHOOL OF ELECTRONICS
VELLORE INSTITUTE OF TECHNOLOGY, VELLORE

Full Mark: 50 Time: (60+ 10) Mins

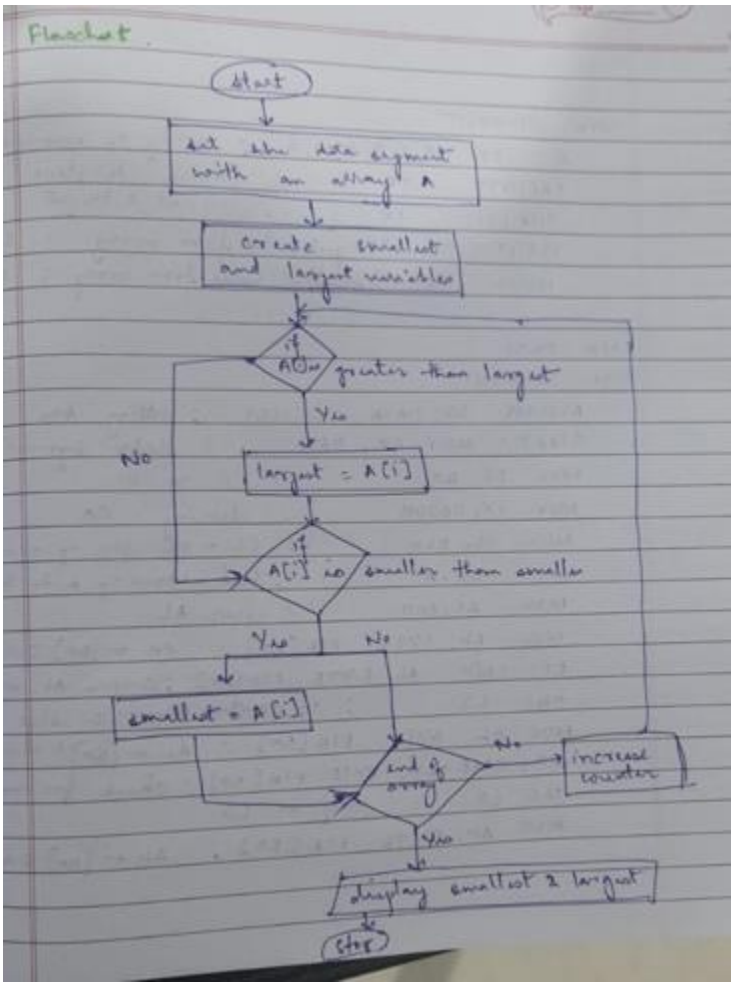
1. **Write an 8086 ALP which is used to find the largest number and smallest number using DOS display interrupts.**

N. B. Everything should be handwritten. Write
a. Complete aim of the experiment
b. Software used, Steps/flowchart
c. Program with detailed comments for every single statement
d. Screenshot of output (file path must contain your name/regn no.)
e. Conclusion

Aim and Software Specifications :



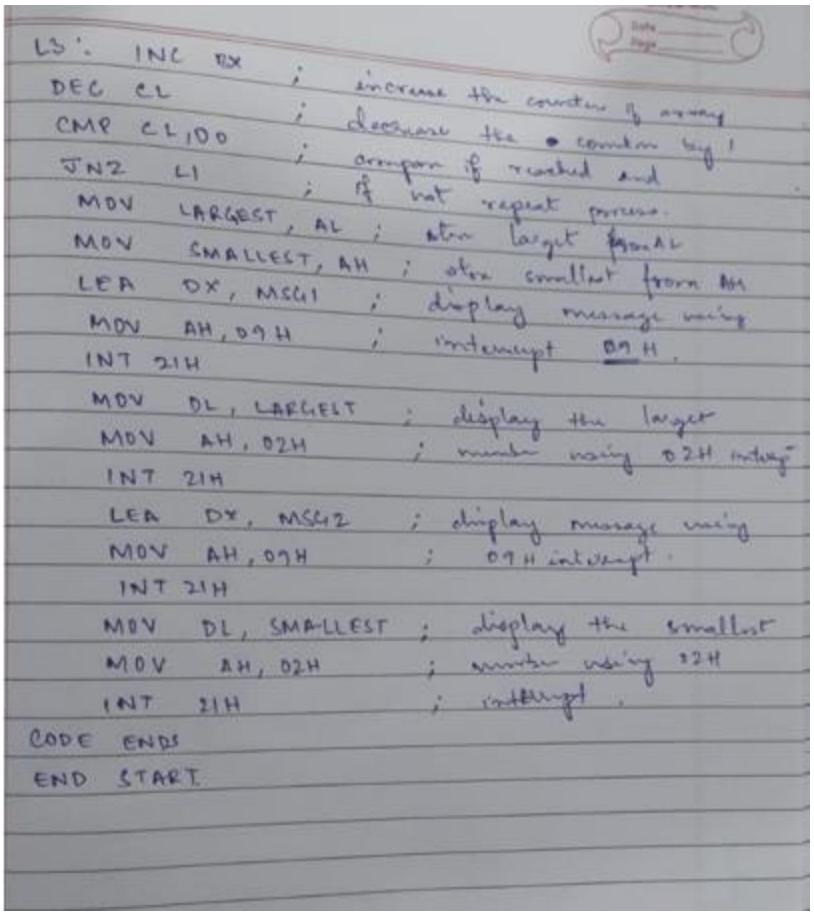
Flowchart :



Handwritten Program :

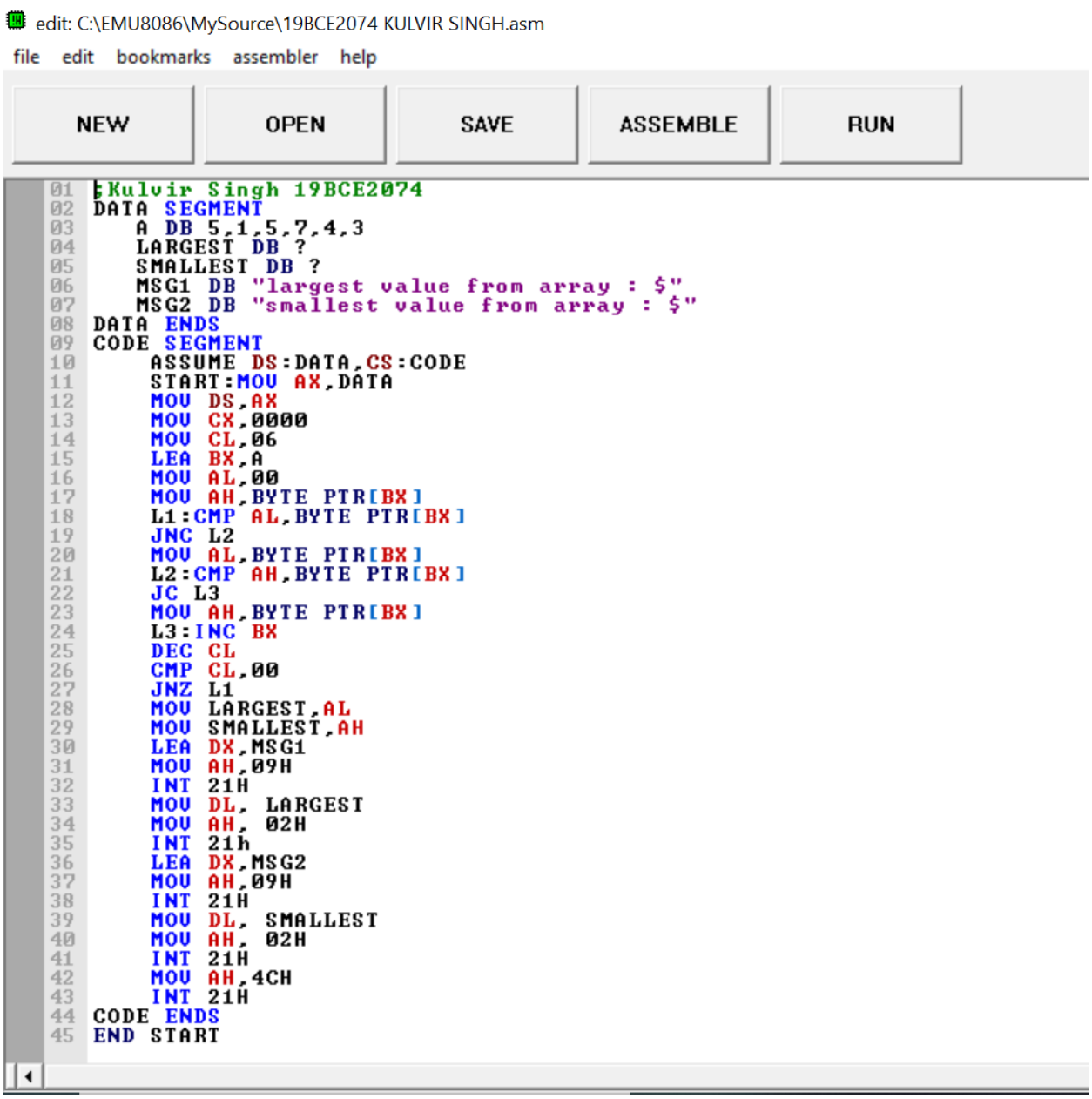
Code

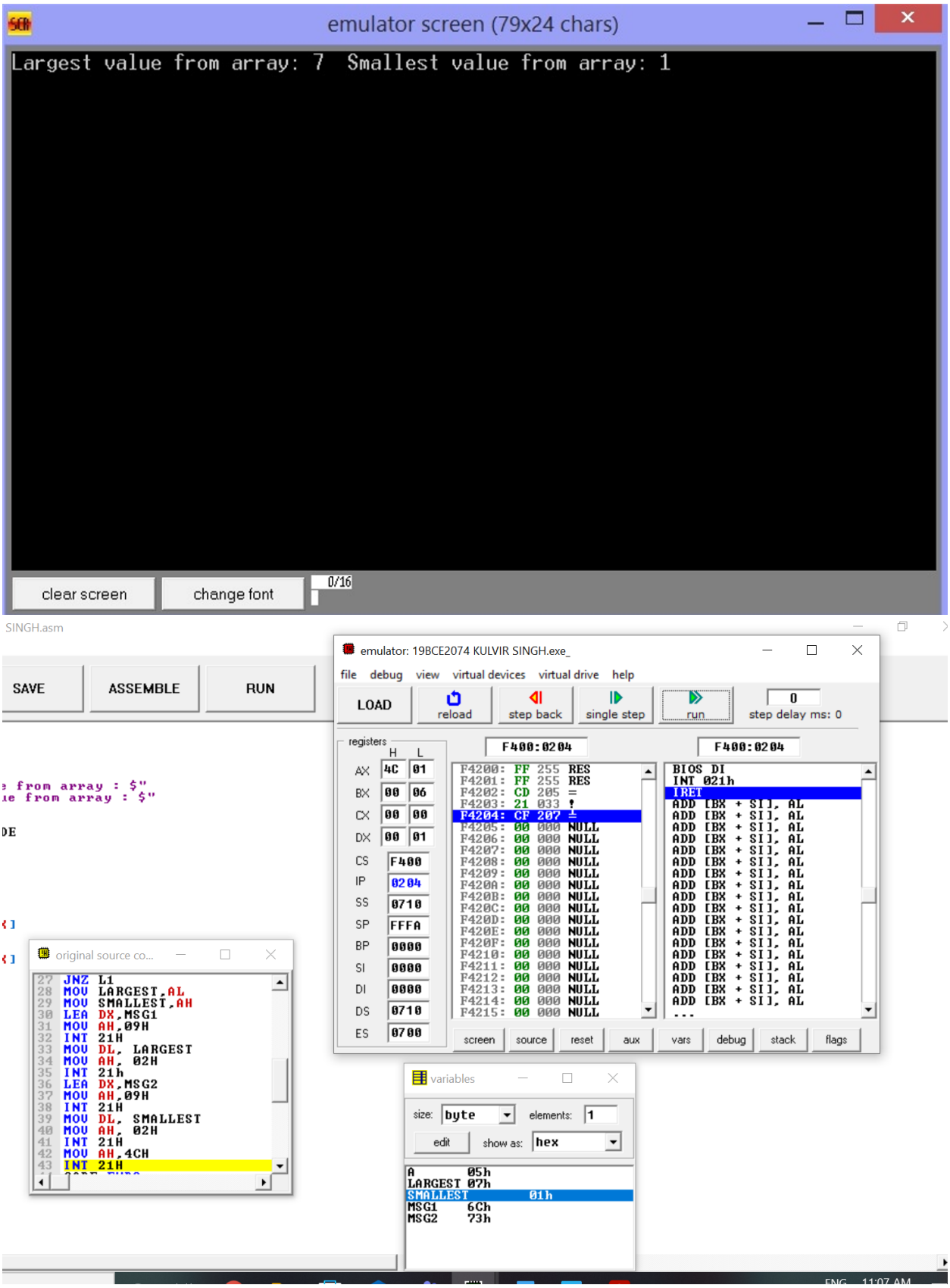
```
DATA SEGMENT
    A DB 5,1,5,7,4,3 ; array of numbers
    LARGEST DB ? ; variable to store
    SMALLEST DB ? ; smallest & largest
    MSG1 DB "Largest value from array : $"
    MSG2 DB "Smallest value from array : $"

DATA ENDS
CODE SEGMENT
    ASSUME DS: DATA, CS: CODE ; setting the
    START: MOV AX, DATA ; data segment
    MOV DS, AX ; to DS.
    MOV CX, 0000H ; clearing CX
    MOV CL, 06H ; CL <- 06 size of array
    LEA BX, A ; load address of A to BX
    MOV AL, 00H ; clear AL
    MOV AH, BYTE PTR [BX] ; AH <- [BX] => data
    L1: CMP AL, BYTE PTR[BX] ; compare AL, array[0]
    JNC L2 ; if greater than L2 else
    MOV AL, BYTE PTR [BX] ; AL <- [BX] => data
    L2: CMP AH, BYTE PTR[BX] ; check for smallest
    JC L3 ; jump to L3
    MOV AH, BYTE PTR [BX] ; AH <- [BX] => data
```



```
L3:  INC BX          ; increase the counter of array
     DEC CL          ; decrease the counter by 1
     CMP CL,00       ; compare if reached end
     JNZ L1          ; if not repeat process.
     MOV LARGEST, AL ; store largest from AL
     MOV SMALLEST, AH ; store smallest from AH
     LEA DX, MSG1    ; display message using
     MOV AH, 09H     ; interrupt 09H.
     INT 21H
     MOV DL, LARGEST ; display the larger
     MOV AH, 02H     ; number using 02H interrupt
     INT 21H
     LEA DX, MSG2    ; display message using
     MOV AH, 09H     ; 09H interrupt.
     INT 21H
     MOV DL, SMALLEST ; display the smallest
     MOV AH, 02H     ; number using 02H
     INT 21H         ; interrupt.
CODE ENDS
END START
```

Screenshots :


edit: C:\EMU8086\MySource\19BCE2074 KULVIR SINGH.asm
file edit bookmarks assembler help
NEW
OPEN
SAVE
ASSEMBLE
RUN
;Kulvir Singh 19BCE2074
DATA SEGMENT
A DB 5,1,5,7,4,3
LARGEST DB ?
SMALLEST DB ?
MSG1 DB "largest value from array : $"
MSG2 DB "smallest value from array : $"
DATA ENDS
CODE SEGMENT
ASSUME DS:DATA,CS:CODE
START:MOV AX,DATA
MOV DS,AX
MOV CX,0000
MOV CL,06
LEA BX,A
MOV AL,00
MOV AH,BYTE PTR[BX]
L1:CMP AL,BYTE PTR[BX]
JNC L2
MOV AL,BYTE PTR[BX]
L2:CMP AH,BYTE PTR[BX]
JC L3
MOV AH,BYTE PTR[BX]
L3:INC BX
DEC CL
CMP CL,00
JNZ L1
MOV LARGEST,AL
MOV SMALLEST,AH
LEA DX,MSG1
MOV AH,09H
INT 21H
MOV DL, LARGEST
MOV AH, 02H
INT 21h
LEA DX,MSG2
MOV AH,09H
INT 21H
MOV DL, SMALLEST
MOV AH, 02H
INT 21H
MOV AH,4CH
INT 21H
CODE ENDS
END START


emulator screen (79x24 chars)
Largest value from array: 7 Smallest value from array: 1
clear screen
change font
0/16
SINGH.asm
SAVE
ASSEMBLE
RUN
emulator: 19BCE2074 KULVIR SINGH.exe_
file debug view virtual devices virtual drive help
LOAD
reload
step back
single step
run
step delay ms: 0
registers
AX 4C 01
BX 00 06
CX 00 00
DX 00 01
CS F400
IP 0204
SS 0710
SP FFFA
BP 0000
SI 0000
DI 0000
DS 0710
ES 0700
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ╧
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
F400:0204
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
screen source reset aux vars debug stack flags
original source co...
27 JNZ L1
28 MOV LARGEST,AL
29 MOV SMALLEST,AH
30 LEA DX,MSG1
31 MOV AH,09H
32 INT 21H
33 MOV DL, LARGEST
34 MOV AH, 02H
35 INT 21h
36 LEA DX,MSG2
37 MOV AH,09H
38 INT 21H
39 MOV DL, SMALLEST
40 MOV AH, 02H
41 INT 21H
42 MOV AH,4CH
43 INT 21H
variables
size: byte
elements: 1
edit
show as: hex
A 05h
LARGEST 07h
SMALLEST 01h
MSG1 6Ch
MSG2 73h

Conclusion :

The program can successfully find out the smallest and the largest number from the array as visible in the output terminal