Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

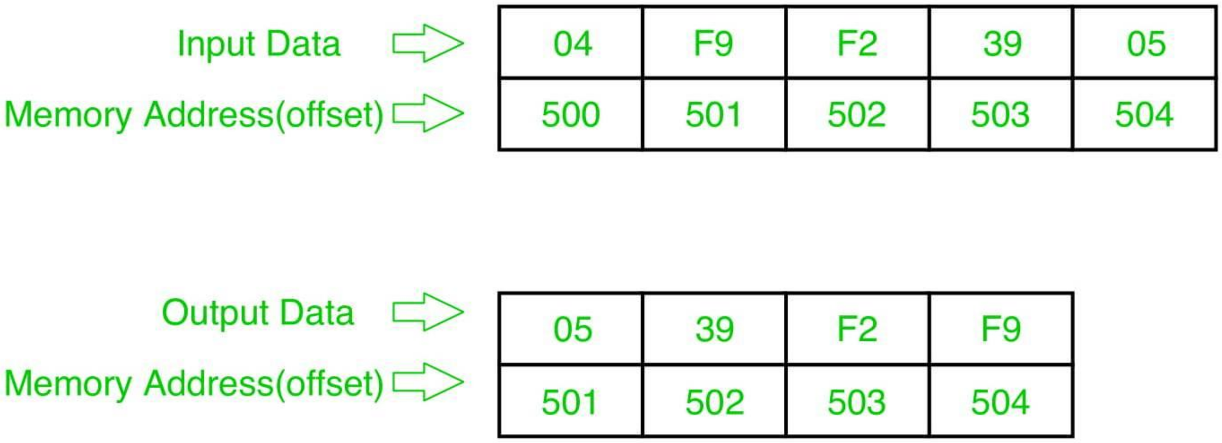
Artificial Intelligence and Data Science Department.
MP / Even Sem 2021-22 / Experiment 7.

YASH SARANG.
47 / D6AD.
EXPERIMENT - 7.

AIM: Assembly program to sort numbers in ascending/ descending order.

THEORY:
Problem – Write a program in 8086 microprocessor to sort numbers in ascending order in an array of n numbers, where size “n” is stored at memory address 2000 : 500 and the numbers are stored from memory address 2000 : 501.

Example –



Input Data ⇨	04	F9	F2	39	05
Memory Address(offset) ⇨	500	501	502	503	504

Output Data ⇨	05	39	F2	F9
Memory Address(offset) ⇨	501	502	503	504

Example explanation:

Pass-1:
F9 F2 39 05
F2 F9 39 05
F2 39 F9 05
F2 39 05 F9 (1 number got fix)

Pass-2:
F2 39 05 F9
39 F2 05 F9
39 05 F2 F9 (2 number got fix)

Pass-3:
39 05 F2 F9
05 39 F2 F9 (sorted)
Algorithm –

1. Load data from offset 500 to register CL (for the count).
2. Travel from starting memory location to last and compare two numbers if the first number is greater than the second number then swap them.
3. First-pass fix the position for the last number.
4. Decrease the count by 1.
5. Again travel from starting memory location to (last-1, with the help of count) and compare two numbers if the first number is greater than the second number then swap them.
6. The second pass fixes the position for the last two numbers.
7. Repeated.

Program 1:

MEMORY ADDRESS	MNEMONICS	COMMENT
400	MOV SI, 500	SI<-500
403	MOV CL, [SI]	CL<-[SI]
405	DEC CL	CL<-CL-1
407	MOV SI, 500	SI<-500
40A	MOV CH, [SI]	CH<-[SI]
40C	DEC CH	CH<-CH-1
40E	INC SI	SI<-SI+1
40F	MOV AL, [SI]	AL<-[SI]
411	INC SI	SI<-SI+1
412	CMP AL, [SI]	AL-[SI]
414	JC 41C	JUMP TO 41C IF CY=1
416	XCHG AL, [SI]	SWAP AL AND [SI]
418	DEC SI	SI<-SI-1
419	XCHG AL, [SI]	SWAP AL AND [SI]
41B	INC SI	SI<-SI+1
41C	DEC CH	CH<-CH-1

41E	JNZ 40F	JUMP TO 40F IF ZF=0
420	DEC CL	CL<-CL-1
422	JNZ 407	JUMP TO 407 IF ZF=0
424	HLT	END

Explanation –

1. MOV SI, 500: set the value of SI to 500.
2. MOV CL, [SI]: load data from offset SI to register CL.
3. DEC CL: decrease the value of register CL BY 1.
4. MOV SI, 500: set the value of SI to 500.
5. MOV CH, [SI]: load data from offset SI to register CH.
6. DEC CH: decrease the value of register CH BY 1.
7. INC SI: increase value of SI BY 1.
8. MOV AL, [SI]: load value from offset SI to register AL.
9. INC SI: increase the value of SI BY 1.
10. CMP AL, [SI]: compares value of register AL and [SI] (AL-[SI]).
11. JC 41C: jump to address 41C if carry generated.
12. XCHG AL, [SI]: exchange the contents of register AL and SI.
13. DEC SI: decrease the value of SI by 1.
14. XCHG AL, [SI]: exchange the contents of register AL and SI.
15. INC SI: increase the value of SI by 1.
16. DEC CH: decrease the value of register CH by 1.
17. JNZ 40F: jump to address 40F if zero flat resets.
18. DEC CL: decrease the value of register CL by 1.
19. JNZ 407: jump to address 407 if zero flat resets.
20. HLT: stop.