

ASCENDING ORDER

EXP NO: 12

AIM: To compute ascending order of an array using 8085 processor.

ALGORITHM:

- 1) Initialize HL pair as memory pointer.
- 2) Get the count at memory and load it into C register
- 3) Copy it in D register (for bubble sort (N-1)) times required.
- 4) Get the first value in A register.
- 5) Compare it with the value at next location.
- 6) If they are out of order, exchange the contents of A register and memory.
- 7) Decrement D register content by 1
- 8) Repeat step 5 and 7 till the value in D register become zero.
- 9) Decrement the C register content by 1.
- 10) Repeat steps 3 to 9 till the value in C register becomes zero.

PROGRAM:

```
LOOP: LXI H,3500
MVI D,00
MVI C,05
LOOP1: MOV A,M
INX H
CMP M
JC LOOP2
MOV B,M
MOV M,A
DCX H
MOV M,B
INX H
MVI D,01
LOOP2: DCR C
JNZ LOOP1
MOV A,D
RRC
JC LOOP
HLT
```

INPUT & OUTPUT

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers: A 00, BC 00 00, DE 00 00, HL 0D B1, PSW 00 00, PC 42 1E, SP FF FF, Int-Reg 00. Flag: S 0, Z 1, AC 0, P 1, C 0.

Decimal - Hex Conversion: Decimal 0, Hex 0. Buttons: To Hex, To Dec.

I/O Ports: 0, -, +, 00. Update Port Value.

Memory: 0, -, +, 00. Update Memory.

Load me at: 1 LOOP: LXI H, 3500, 2 MVI D, 00, 3 MVI C, 05, 4 LOOP1: MOV A, M, 5 INX H, 6 CMP M, 7 JC LOOP2, 8 MOV B, M, 9 MOV M, A, 10 DCX H, 11 MOV M, B, 12 INX H, 13 MVI D, 01, 14 LOOP2: DCR C, 15 JNZ LOOP1, 16 MOV A, D, 17 RRC, 18 JC LOOP, 19 HLT.

Start: 3500. OK.

Address (Hex)	Address	Data
0DAC	3500	2
0DAD	3501	8
0DAE	3502	12
0DAF	3503	20
0DB0	3504	23
0DB1	3505	35
0DB2	3506	34
0DB3	3507	0
0DB4	3508	0
0DB5	3509	0
0DB6	3510	0
0DB7	3511	0
0DB8	3512	0
0DB9	3513	0

Line No Assembler Message
0 Program assembled successfully

Simulator: Idle

RESULT: Thus the program was executed successfully using 8085 processor simulator.