

DESCENDING ORDER

EXP NO: 13

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

ALGORITHM:

- 1) Initialise the HL pair as a 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 the 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 steps 5 and 7 till the value in D register becomes 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

JNC 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:

Start	3500	OK
Address (Hex)	Address	Data
0DAC	3500	4
0DAD	3501	89
0DAE	3502	12
0DAF	3503	52
0DB0	3504	9

OUTPUT:

GNUSim8085 - 8085 Microprocessor Simulator

File Reset Assembler Debug Help

Registers: A 00, BC 34 00, DE 00 00, HL 00 B1, PSW 00 00, PC 42 22, SP FF FF, Int-Reg 00. Flag: S 0, Z 1, AC 0, P 1, C 0.

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

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

Memory: 0, 00. Update Memory.

Load me at: 1 <Program title>, 2 jmp start, 3 rdata, 4 rcode, 5 start: pop, 6 LOOP: LXI H,3500, 7 MVI D,00, 8 MVI C,05, 9 LOOP1: MOV A,M, 10 INX H, 11 CMP M, 12 JNC LOOP2, 13 MOV B,M, 14 MOV M,A, 15 DCX B, 16 MOV M,B, 17 INX B, 18 MVI D,01, 19 LOOP2: DCR C, 20 JNZ LOOP1, 21 MOV A,D, 22 RRC, 23 JC LOOP, 24 hit.

Start 3500 OK

Address (Hex)	Address	Data
0DAC	3500	89
0DAD	3501	52
0DAE	3502	12
0DAF	3503	9
0DB0	3504	4
0DB1	3505	0
0DB2	3506	0
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

31°C Feels hotter

Search

ENG IN 09:57 17-10-2023

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