

Selection Sort

// FIVE numbers stored from 2000H are sorted in ascending order using Selection Sort

START: MVI C,04 // Counter

LXI SP,4000 // We are initializing SP to store memory address of the number that is found to be smallest

LXI H,2000 // Initializing memory location

W: MOV B,L // Saving memory location of the number in consideration

MOV E,M // Making replica of the element in memory

MOV D,C // Initializing inner loop counter

MOV A,M // Moving the number in Accumulator to get it compared

LOOP: INX H // Incrementing HL pair by 1

CMP M // Comparing with the element in new memory location

JNC SWAP // If new number is found small then going to SWAP

DCR D // Decrementing D

JZ SWAP_WITH_MEMORY // If D = 0 --> Going to SWAP_WITH_MEMORY

JNZ LOOP // If D!=0 going to Loop

SWAP: MOV A,M // Moving the so found smallest number in Accumulator

PUSH H // Saving it's memory in SP

DCR D // Decrementing D

JNZ LOOP // If D!=0 moving to LOOP

SWAP_WITH_MEMORY: POP H // Popping SP' content to HL

MOV M,E // Moving content of E to M

MOV L,B // Moving Content of B to L

MOV M,A // Moving Content of A to M

INX H // Incrementing H

DCR C // Decrementing C

JNZ W // If C!=0 --> Moving to W

HLT // STOP

ORG 2000

DB 12H,01H,03H,02H,04H

// SELECTION SORT

// FIVE numbers stored from 2000H are sorted in ascending order using Selection Sort

START: MVI C,04 // Counter
 LXI SP,4000 // We are initializing SP to store memory
 address of the number that is found to be smallest
 LXI H,2000 // Initializing memory location

W: MOV B,L // Saving memory location of the number in
 consideration
 MOV E,M // Making replica of the element in memory
 MOV D,C // Initializing inner loop counter
 MOV A,M // Moving the number in Accumulator to get it
 compared

LOOP: INX H // Incrementing HL pair by 1
 CMP M // Comparing with the element in new memory
 location
 JNC SWAP // If new number is found small then going to
 SWAP
 DCR D // Decrementing D
 JZ SWAP_WITH_MEMORY // If D = 0 --> Going to
 SWAP_WITH_MEMORY
 JNZ LOOP // If D!=0 going to Loop

SWAP: MOV A,M // Moving the so found smallest number in
 Accumulator
 PUSH H // Saving it's memory in SP
 DCR D // Decrementing D
 JNZ LOOP // If D!=0 moving to LOOP

SWAP_WITH_MEMORY: POP H // Popping SP' content to HL
 MOV M,E // Moving content of E to M
 MOV L,B // Moving Content of B to L
 MOV M,A // Moving Content of A to M
 INX H // Incrementing H
 DCR C // Decrementing C
 JNZ W // If C!=0 --> Moving to W
 HLT // STOP

ORG 2000

DB 12H,01H,03H,02H,04H

Memory Editor	
Memory Range: 0000 --- FFFF	
Memory Address	Value
001E	E1
001F	73
0020	68
0021	77
0022	23
0023	0D
0024	C2
0025	08
0027	76
2000	12
2001	01
2002	03
2003	02
2004	04

Swapping first smallest number

Memory Editor	
Memory Range: 0000 --- FFFF	
Memory Address	Value
001E	E1
001F	73
0020	68
0021	77
0022	23
0023	0D
0024	C2
0025	08
0027	76
2000	01
2001	12
2002	03
2003	02
2004	04
3FFE	01
3FFF	20

Swapping second smallest number

Memory Editor	
Memory Range: 0000 --- FFFF	
Memory Address	Value
001E	E1
001F	73
0020	68
0021	77
0022	23
0023	0D
0024	C2
0025	08
0027	76
2000	01
2001	02
2002	03
2003	12
2004	04
3FFC	03
3FFD	20
3FFE	02
3FFF	20

Swapping third smallest number

Memory Editor	
Memory Range: 0000 --- FFFF	
Memory Address	Value
001E	E1
001F	73
0020	68
0021	77
0022	23
0023	0D
0024	C2
0025	08
0027	76
2000	01
2001	02
2002	03
2003	04
2004	12
3FFC	03
3FFD	20
3FFE	04
3FFF	20

Swapping fourth smallest number

Registers :									
Register	Value	7	6	5	4	3	2	1	0
Accumulator	04	0	0	0	0	0	1	0	0
Register B	03	0	0	0	0	0	0	1	1
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	12	0	0	0	1	0	0	1	0
Register H	20	0	0	1	0	0	0	0	0
Register L	04	0	0	0	0	0	1	0	0
Memory(M)	12	0	0	0	1	0	0	1	0
Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Register	54	0	1	0	1	0	1	0	0
Type	Value								
Stack Pointer(SP)	4000								
Memory Pointer (HL)	2004								
Program Status Word(PSW)	0454								
Program Counter(PC)	0027								
Clock Cycle Counter	760								
Instruction Counter	110								

Final values in the register pair