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 // Poping 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

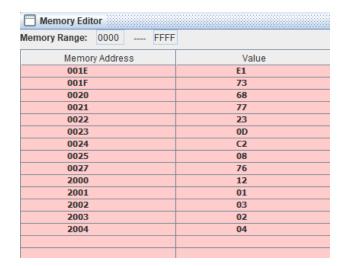
```
JNZ W // If C!=0 --> Moving to W
        HLT // STOP
# ORG 2000
# DB 12H,01H,03H,02H,04H
      // SELECTION SORT
      // FIVE numbers stored from 2000H are sorted in ascending order using
      Selection Sort
      START:
                    MVI C,04 // Counter
                    LXI SP,4000 // 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
                    MOV A,M // Moving the so found smallest number in
      SWAP:
      Accumulator
                    PUSH H // Saving it's memory in SP
                    DCR D // Decrementing D
                    JNZ LOOP // If D!=0 moving to LOOP
                                POP H // Poping SP' content to HL
      SWAP WITH MEMORY:
                    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

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

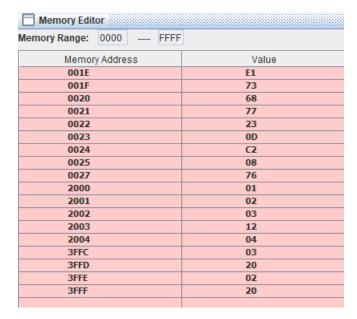
ORG 2000



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 first smallest number

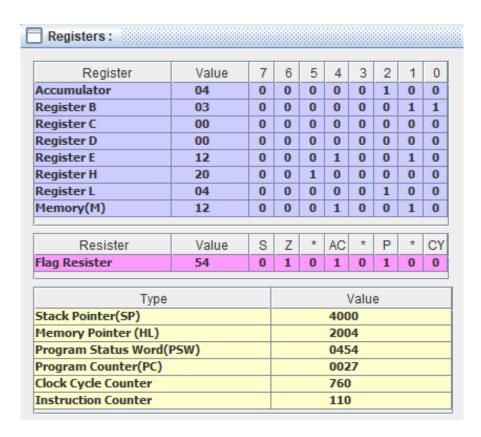
Swapping second smallest number



Memory Editor				
Memory Range: 0	000	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 third smallest number

Swapping fourth smallest number



Final values in the register pair