# PROGRAM 5: SORTING (ascending)

**INTRO:** To write a program to perform sorting in ascending order using 8085(bubble sorting).

**FLOW CHART:**

Jump if not zero

END

Jump if not zero

Decrement the value of B

Increment H & decrement C

Move the contents from D to M

Decrement address of HL pair

Move M to D & then from Acc. to M

Jump IF Carry

Increment the address of H-L pair & compare

Move the value from M to A

Load the address to H-L pair

Assign values to B & C registers

Start

|  |  |  |
| --- | --- | --- |
| ADDRESS | MNEMONICS | EXPLANATION |
| 4100 | MVIB,05 | Move immediately value ‘05’ in hexadecimal to initialize counter 1 |
| 4102(loop 3) | MVIC,05 | Move immediately value ‘05’ in hexadecimal to initialize counter 2 |
| 4104 | LXIH,4200 | Set the memory pointer(HL pair) to address 4200 |
| 4107(loop2) | MOV A,M | Move the value of M to Accumulator |
| 4108 | INX H | Increment memory pointer (HL pair) |
| 4109 | CMP M | Compare number with next number |
| 410A(loop1) | JC 4115 | If carry (if no. Is greater) then skip to 4115 |
| 410D | JZ 4115 | If zero(if number is equal) then also skip to 4115 |
| 4110 | MOV D,M | Move value of M to register D |
| 4111 | MOV MA | Move value of accumulator to M |
| 4112 | DCX H | Decrement memory pointer |
| 4113 | MOV M,D | Move value of register D to M |
| 4114 | INX H | Increment memory pointer(hl pair) |
| 4115(loop 1) | DCR C | Decrement value in counter 2(register C) |
| 4116(loop 2) | JNZ 4107 | Jump if C is not zero |
| 4119 | DCR B | Decrement value in counter 1 (register 2) |
| 411A(loop 3) | JNZ 4102 | Jump if B(counter 1) is not zero |
| 411D | HLT | Terminate /halt |

**PROGRAMING CODE:**

**MODEL CALCULATIONS:**

Ex-

Input no:

3 (gets compared to next no, and is interchanged if its greater)’ 2

The process continues to happen till ‘2’ reaches the top and with each comparison the value of counter 2 is decreased by 1

2 3

4 4

6 5

5 6

Similarly the process continues for the numbers now obtained and it happens till all numbers are sorted

Output:

2

3

4

5

6

END