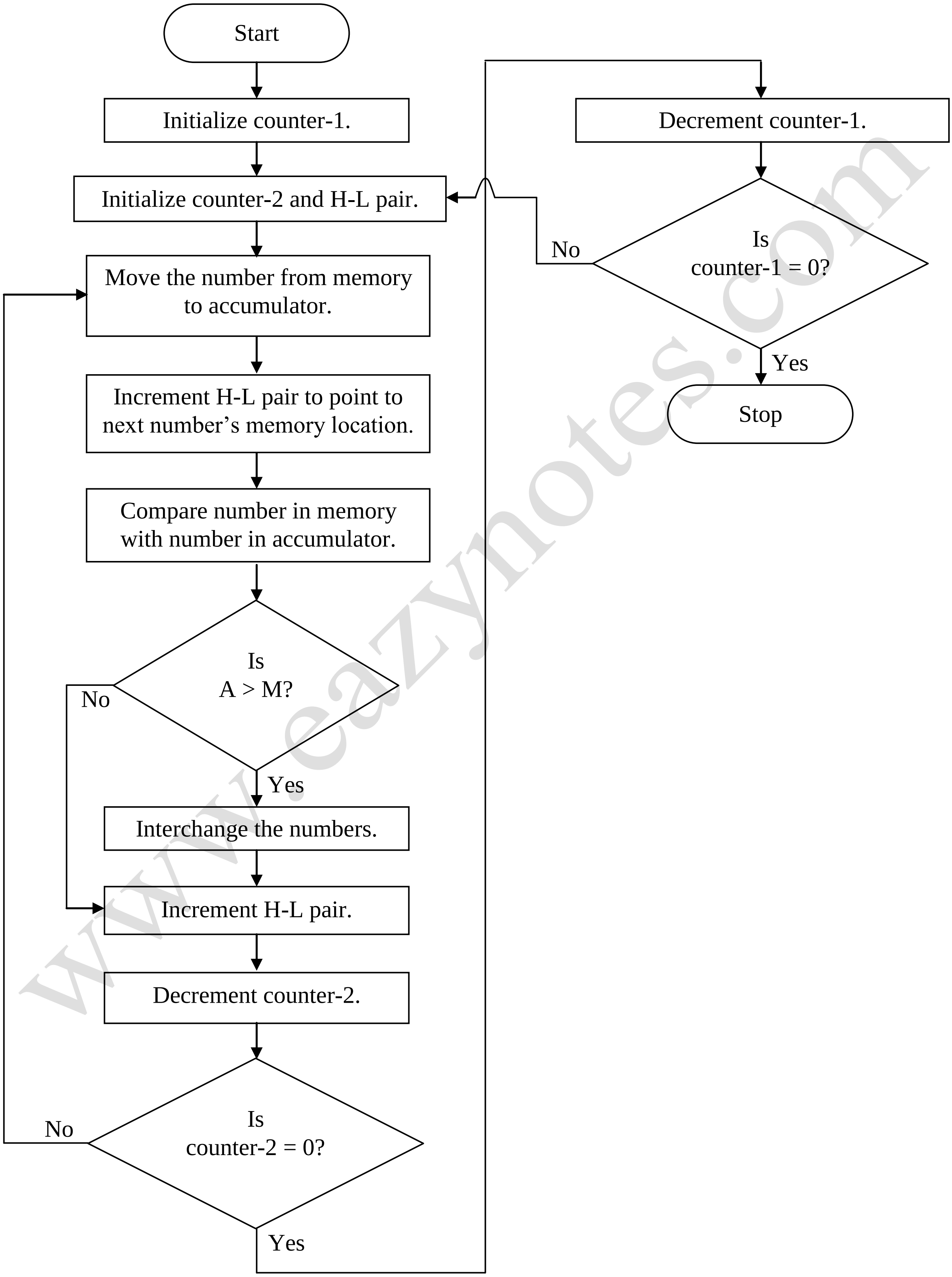
Program 25: Sort the array in ascending order.

Flowchart:


Start
Initialize counter-1.
Initialize counter-2 and H-L pair.
Move the number from memory to accumulator.
Increment H-L pair to point to next number's memory location.
Compare number in memory with number in accumulator.
Is A > M?
No
Yes
Interchange the numbers.
Increment H-L pair.
Decrement counter-2.
Is counter-2 = 0?
No
Yes
Decrement counter-1.
Is counter-1 = 0?
No
Yes
Stop

Program:

Address	Mnemonics	Operand	Opcode	Remarks
2000	MVI	B, 05H	06	Initialize counter-1.
2001			05	Immediate value 05H.
2002	MVI	C, 05H	0E	Initialize counter-2.
2003			05	Immediate value 05H.
2004	LXI	H, 3000H	21	Load H-L pair with address 3000H.
2005			00	Lower-order of 3000H.
2006			30	Higher-order of 3000H.
2007	MOV	A, M	7E	Move the number from memory to reg. A.
2008	INX	H	23	Increment H-L pair.
2009	CMP	M	BD	Compare the number with next number.
200A	JC	2015H	DA	Don’t interchange if number < next number.
200B			15	Lower-order of 2015H.
200C			20	Higher-order of 2015H.
200D	JZ	2015H	CA	Don’t interchange if number = next number.
200E			15	Lower-order of 2015H.
200F			20	Higher-order of 2015H.
2010	MOV	D, M	56	Otherwise, swap the numbers. Move next number from memory to D.
2011	MOV	M, A	77	Move first number from A to memory.
2012	DCX	H	2B	Decrement H-L pair.
2013	MOV	M, D	72	Move next number from D to memory.
2014	INX	H	23	Increment H-L pair.
2015	DCR	C	0D	Decrement counter 2.
2016	JNZ	2007H	C2	If counter-2 ≠ 0, repeat.
2017			07	Lower-order of 2007H.
2018			20	Higher-order of 2007H.
2019	DCR	B	05	Decrement counter-1.
201A	JNZ	2002	C2	If counter-1 ≠ 0, repeat.
201B			02	Lower-order of 2002H.
201C			20	Higher-order of 2002H.
201D	HLT		76	Halt.

Explanation:

- This program sorts an array in ascending order.
- Let us assume that there are five numbers in the array and its starting address is 3000H.
- Initially, counter-1 and counter-2 are initialized with the size of the array.
- H-L pair is pointed to the starting address of the array.
- In the first iteration, first number is compared with the second number.
- If first number < second number, then do not interchange them. Otherwise, if first number > second number, then swap them.
- In the next iteration, first number is compared with the third number.
- If first number < third number, then do not interchange them. Otherwise, if first number > third number, then swap them.
- In the next iteration, first number is compared with the fourth number and the process continues until counter-2 becomes zero.
- When counter-2 becomes zero, counter-1 is decremented and the process continues until all the numbers are arranged in ascending order.

Output:

Before Execution:		**After Execution:**	
3000H:	05H	3000H:	01H
3001H:	15H	3001H:	05H
3002H:	01H	3002H:	15H
3003H:	65H	3003H:	32H
3004H:	32H	3004H:	65H