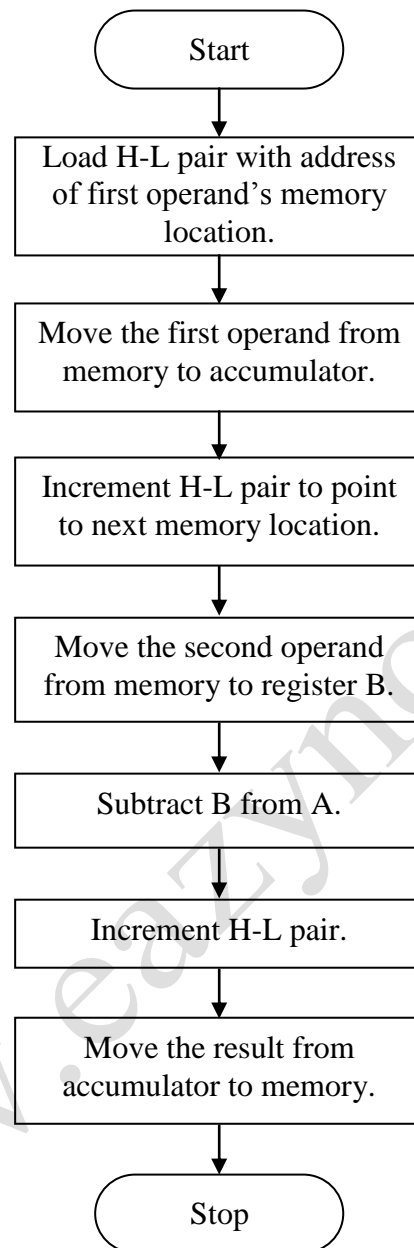


**Program 11:** Subtract two 8-bit numbers without considering the borrow.

**Flowchart:**



**Program:**

Address	Mnemonics	Operand	Opcode	Remarks
2000	LXI	H, 3000H	21	Load H-L pair with address 3000H.
2001			00	Lower-order of 3000H.
2002			30	Higher-order of 3000H.
2003	MOV	A, M	7E	Move the 1 <sup>st</sup> operand from memory to reg. A.
2004	INX	H	23	Increment H-L pair.
2005	MOV	B, M	46	Move the 2 <sup>nd</sup> operand from memory to reg. B.
2006	SUB	B	90	Subtract B from A.
2007	INX	H	23	Increment H-L pair.
2008	MOV	M, A	77	Move the result from reg. A to memory.
2009	HLT		76	Halt.

**Explanation:**

- This program subtracts two operands stored in memory location 3000H and 3001H, without considering the borrow taken (if any).
- Let us assume that the operands stored at memory location 3000H is 05H and 3001H is 02H.
- Initially, H-L pair is loaded with the address of first memory location.
- The first operand is moved to accumulator from memory location 3000H and H-L pair is incremented to point to next memory location.
- The second operand is moved to register B from memory location 3001H.
- The two operands are subtracted and the result is stored in accumulator.
- H-L pair is again incremented and the result is moved from accumulator to memory location 3002H.

**Output:****Before Execution:**

3000H: 05H  
3001H: 02H

**After Execution:**

3002H: 03H