Program 18: Largest of two 8-bit numbers.

Flowchart:

```
                          ┌─────────┐
                          │  Start  │
                          └─────────┘
                               │
                               ▼
              ┌────────────────────────────┐        ┌────────────────────────────┐
              │ Load H-L pair with address │        │  Move the result from      │
              │ of first operand's memory  │        │  accumulator to memory.    │
              │          location.         │        └────────────────────────────┘
              └────────────────────────────┘                     │
                               │                                 ▼
                               ▼                           ┌─────────┐
              ┌────────────────────────────┐               │  Stop   │
              │ Move the first operand from│               └─────────┘
              │  memory to accumulator.    │
              └────────────────────────────┘
                               │
                               ▼
              ┌────────────────────────────┐
              │ Increment H-L pair to point│
              │  to next memory location.  │
              └────────────────────────────┘
                               │
                               ▼
              ┌────────────────────────────┐
              │  Move the second operand   │
              │  from memory to register B.│
              └────────────────────────────┘
                               │
                               ▼
              ┌────────────────────────────┐
              │     Compare B with A.      │
              └────────────────────────────┘
                               │
                               ▼
                        ╱──────────╲
                  No   ╱     If      ╲
              ◄───────◄    Carry?     ►
                        ╲            ╱
                         ╲──────────╱
                             │ Yes
                             ▼
              ┌────────────────────────────┐
              │ Move data from register B  │
              │     to accumulator.        │
              └────────────────────────────┘
                             │
                             ▼
              ┌────────────────────────────┐
              │    Increment H-L pair.     │
              └────────────────────────────┘
```
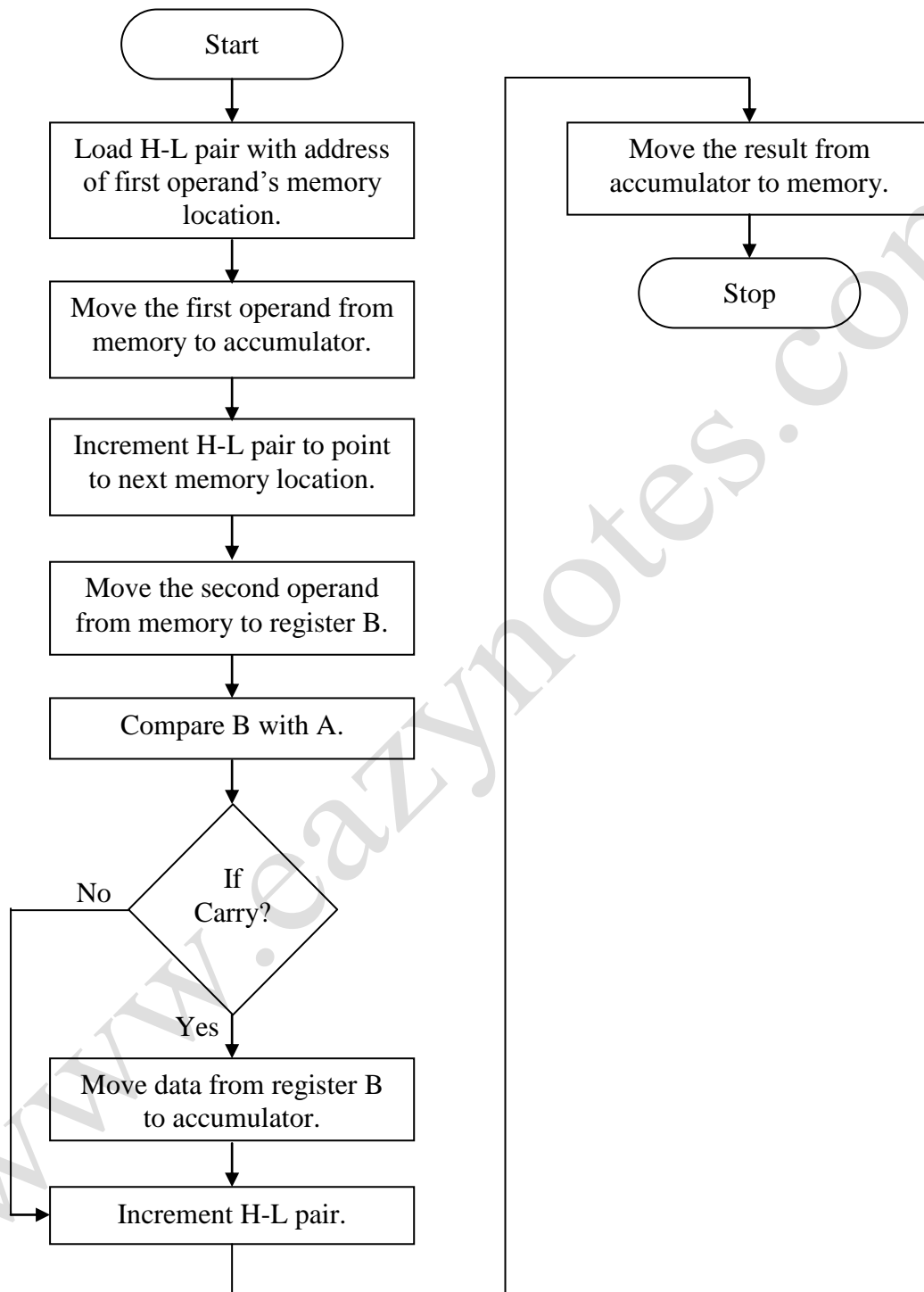
Program:

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 1st operand from memory to reg. A.
2004	INX	H	23	Increment H-L pair.
2005	MOV	B, M	46	Move the 2nd operand from memory to reg. B.
2006	CMP	B	B8	Compare B with A.
2007	JNC	200BH	D2	Jump to address 200BH if there is no carry.
2008			0B	Lower-order of 200BH.
2009			20	Higher-order of 200BH.
200A	MOV	A, B	78	Move largest from reg. B to reg. A.
200B	INX	H	23	Increment H-L pair.
200C	MOV	M, A	77	Move the result from reg. A to memory.
200D	HLT		76	Halt.

Explanation:

- This program compares the two operands to find the largest out of them.

- After comparison, the largest of two must be in accumulator. If it is already in accumulator, then it is moved to memory.

- If it is not in accumulator, then first it is moved to accumulator and then from there, it is moved to memory.

- Let us assume that the operands stored at memory location 3000H is 25H and 3001H is 15H.

- 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 compared.

- After comparison, if A > B, then CF = 0, and if A < B, then CF = 1.

- Carry flag is checked for carry. If there is a carry, it means B is greater than A and it is moved to accumulator.

- At last, H-L pair is incremented and the largest number is moved from accumulator to memory location 3002H.

Output:

Before Execution:

       3000H: 25H

       3001H: 15H

After Execution:

       3002H: 25H