**Practical No. 9**

**Title:** Finding the Largest Number in an Array of Ten 8-bit Numbers

**Objective:** To determine the largest number in an array of ten 8-bit numbers using a microprocessor/microcontroller.

**Theory:**

Finding the largest number in an array involves **iterative comparison**. The process follows these steps:

1. Assume the **first number** in the array is the largest.
2. Compare it with the next number. If the next number is larger, update the largest number.
3. Repeat this process for all ten numbers.
4. The final value stored in the **largest number** register is the largest number in the array.

**Example (Array of 8-bit Numbers in Binary):**

{01010110, 01101100, 00111011, 11001010, 10010101, 01111100, 10111010, 11100011, 01011001, 01100001}

After comparing each element, the **largest number** is found.

**Materials/Tools Required:**

* Microprocessor/microcontroller (e.g., 8085/8051)
* Assembler/Simulator
* Computer system with programming software
* Binary calculator (optional)

**Procedure:**

1. **Initialize Registers:**
   * Load the first number from the array into the accumulator (A register).
   * Store it as the current largest number.
2. **Compare with Next Element:**
   * Load the next number in a temporary register (e.g., B register).
   * Subtract it from the stored largest number.
   * If the result is negative (Carry Flag set), update the largest number with the new value.
3. **Repeat for All Numbers:**
   * Move to the next number in the array.
   * Continue comparing and updating the largest number.
4. **Store and Display the Largest Number:**
   * Once the loop completes, the largest number is stored in a register or memory.
   * Display the final result.

**Observations:**

* The largest number in the array is correctly identified.
* The Carry Flag (CY) helps in determining whether to update the largest number.
* This method works efficiently with a fixed-size array but can be extended for larger datasets.

**Conclusion:**

The largest number in an array of ten 8-bit numbers is successfully determined using iterative comparisons and conditional updating. This approach is essential in sorting, searching, and data analysis applications.

**Applications:**

* Used in sorting and searching algorithms in digital systems.
* Essential in microprocessor-based data analysis operations.
* Applied in real-time embedded systems for sensor data processing.