- **Purpose**:

The code implements the bubble sort algorithm to sort an array of 16-bit unsigned integers in ascending order.

- **Data Initialization:**

The data section initializes an array named data with 10 elements of 16-bit unsigned integers.
Each element represents a value to be sorted.

- **Swap Flag:**

A byte variable named swap is used as a flag to indicate whether any swaps occurred during a pass through the array.
If no swaps occur during a pass, the array is already sorted, and the sorting process can terminate early.

- **Outer Loop:**

The outer loop iterates through the array elements.
It compares adjacent pairs of elements and swaps them if they are out of order.
The loop continues until no swaps are made in a pass, indicating that the array is sorted.

- **Inner Loop:**

Within the outer loop, an inner loop compares each pair of adjacent elements.
If the current element is greater than the next element, they are swapped, and the swap flag is set to indicate a swap occurred.

- **Bubble Sort Logic:**

Bubble sort repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order.
This process continues until the entire list is sorted.

- **Optimization:**

The swap flag optimization improves the efficiency of the bubble sort algorithm.
If no swaps occur during a pass, the array is already sorted, and the algorithm can terminate early.

This optimization reduces unnecessary iterations through the array.

- Termination:

After the sorting process completes, the program exits by invoking the DOS interrupt int 0x21 with the termination code 0x4c00.