|  |  |
| --- | --- |
| **Bubble sort in C++** | |
| #include <iostream>  using namespace std;  void BubbleSort(int arr[], int n) {  for (int i = 0; i < n - 1; i++) {  for (int j = 0; j < n - 1 - i; j++) {  if (arr[j] > arr[j + 1]) {  // Swap arr[j] and arr[j+1]  int temp = arr[j];  arr[j] = arr[j + 1];  arr[j + 1] = temp;  }  }  }  }  int main() {  int arr[] = {0, 1, 5, 7, 8, 9, 4};  int n = sizeof(arr) / sizeof(arr[0]);  BubbleSort(arr, n);  cout << "Sorted array: ";  for (int i = 0; i < n; i++) {  cout << arr[i] << " ";  }  cout << endl;  return 0;  } | **Dry Run Table:**  **Initial:**  [0, 1, 5, 7, 8, 9, 4]  **Pass 1 (i = 0):**   | **Compare arr[j]** | **Swap?** | **Result** | | --- | --- | --- | | 0 and 1 | No | [0, 1, 5, 7, 8, 9, 4] | | 1 and 5 | No | [0, 1, 5, 7, 8, 9, 4] | | 5 and 7 | No | [0, 1, 5, 7, 8, 9, 4] | | 7 and 8 | No | [0, 1, 5, 7, 8, 9, 4] | | 8 and 9 | No | [0, 1, 5, 7, 8, 9, 4] | | 9 and 4 | **Yes** | [0, 1, 5, 7, 8, 4, 9] |   ✅ Largest element 9 moved to the end.  **Pass 2 (i = 1):**   | **Compare arr[j]** | **Swap?** | **Result** | | --- | --- | --- | | 0 and 1 | No | [0, 1, 5, 7, 8, 4, 9] | | 1 and 5 | No | [0, 1, 5, 7, 8, 4, 9] | | 5 and 7 | No | [0, 1, 5, 7, 8, 4, 9] | | 7 and 8 | No | [0, 1, 5, 7, 8, 4, 9] | | 8 and 4 | **Yes** | [0, 1, 5, 7, 4, 8, 9] |   ✅ Second-largest 8 in place.  **Pass 3 (i = 2):**   | **Compare arr[j]** | **Swap?** | **Result** | | --- | --- | --- | | 0 and 1 | No | [0, 1, 5, 7, 4, 8, 9] | | 1 and 5 | No | [0, 1, 5, 7, 4, 8, 9] | | 5 and 7 | No | [0, 1, 5, 7, 4, 8, 9] | | 7 and 4 | **Yes** | [0, 1, 5, 4, 7, 8, 9] |   **Pass 4 (i = 3):**   | **Compare arr[j]** | **Swap?** | **Result** | | --- | --- | --- | | 0 and 1 | No | [0, 1, 5, 4, 7, 8, 9] | | 1 and 5 | No | [0, 1, 5, 4, 7, 8, 9] | | 5 and 4 | **Yes** | [0, 1, 4, 5, 7, 8, 9] |   **Pass 5 (i = 4):**   | **Compare arr[j]** | **Swap?** | **Result** | | --- | --- | --- | | 0 and 1 | No | [0, 1, 4, 5, 7, 8, 9] | | 1 and 4 | No | [0, 1, 4, 5, 7, 8, 9] |   **Pass 6 (i = 5):**   | **Compare arr[j]** | **Swap?** | **Result** | | --- | --- | --- | | 0 and 1 | No | [0, 1, 4, 5, 7, 8, 9] |   **🏁 Final Sorted Array:**  Sorted array: 0 1 4 5 7 8 9 |
| Sorted array: 0 1 4 5 7 8 9 | |