|  |  |
| --- | --- |
| **Insertion Sort in C++** | |
| #include <iostream>  using namespace std;  // void insertionSort(int arr[], int n) {  // for (int i = 1; i < n; i++)  // {  // int key=arr[i];  // int j=i-1;  // while(j>=0 && arr[j]>key){  // arr[j+1]=arr[j];  // j=j-1;  // }  // arr[j + 1] = key;  // }  // }  void insertionSort(int arr[], int n) {  for (int i = 1; i < n; i++)  {  int j=i;  while(j>0 && arr[j-1]>arr[j]){  swap(arr[j],arr[j-1]);  j--;  }  }  }  int main() {  int arr[] = {12, 11, 13, 5, 6};  int n = sizeof(arr)/sizeof(arr[0]);  insertionSort(arr, n);  cout << "Sorted array: \n";  for(int i = 0; i < n; i++) {  cout << arr[i] << " ";  }  return 0;  } | Input: arr[] = {12, 11, 13, 5, 6} 🧮 Step-by-Step Dry Run (Tabular Form)  | **i (loop index)** | **j (inner loop)** | **Comparison** | **Action** | **Array State** | | --- | --- | --- | --- | --- | | 1 | 1 | 11 < 12 | swap(11, 12) | [11, 12, 13, 5, 6] | | 2 | 2 | 13 < 12? ❌ | no swap | [11, 12, 13, 5, 6] | | 3 | 3 | 5 < 13 → swap | [11, 12, 5, 13, 6] |  | |  | 2 | 5 < 12 → swap | [11, 5, 12, 13, 6] |  | |  | 1 | 5 < 11 → swap | [5, 11, 12, 13, 6] |  | | 4 | 4 | 6 < 13 → swap | [5, 11, 12, 6, 13] |  | |  | 3 | 6 < 12 → swap | [5, 11, 6, 12, 13] |  | |  | 2 | 6 < 11 → swap | [5, 6, 11, 12, 13] |  |  ✅ Final Output: Sorted array:  5 6 11 12 13 |
| 5 6 11 12 13 | |