|  |  |
| --- | --- |
| **Count Sort in C++** | |
| #include <iostream>  #include <cstring>  using namespace std;  string countSort(string s) {  char arr[s.length()];  strcpy(arr, s.c\_str());  char maxch = 'a';  for (int i = 0; i < strlen(arr); i++) {  if (arr[i] > maxch) {  maxch = arr[i];  }  }  int max = maxch - 'a';  int count[max + 1] = {0};  for (int i = 0; i < strlen(arr); i++) {  int val = arr[i] - 'a';  count[val]++;  }  int k = 0;  for (int i = 0; i <= max; i++) {  int c = count[i];  for (int j = 0; j < c; j++) {  arr[k] = i + 'a';  k++;  }  }  string sortedString(arr);  return sortedString;  }  int main() {  string input = "countingsortexample";  string sortedString = countSort(input);  cout << "Original String: " << input << endl;  cout << "Sorted String: " << sortedString << endl;  return 0;  } | Step-by-Step Dry Run:Step 1: Copy string to character array strcpy(arr, s.c\_str());  Now arr = "countingsortexample" Step 2: Find max character (in terms of ASCII) char maxch = 'x'; // max character = 'x'  int max = maxch - 'a'; // max = 23 Step 3: Count frequency of each character  | **Character** | **Count** | | --- | --- | | a | 1 | | c | 1 | | e | 2 | | g | 1 | | i | 1 | | l | 1 | | m | 1 | | n | 2 | | o | 2 | | p | 1 | | r | 1 | | s | 1 | | t | 2 | | u | 1 | | x | 1 |  Step 4: Reconstruct the sorted array Characters are added in order of 'a' to 'x' based on count.  Sorted string becomes:  "aceegilmnnooprsttux" ✅ Output: Original String: countingsortexample  Sorted String: aceegilmnnooprsttux |
| Original String: countingsortexample  Sorted String: aceegilmnnooprsttux | |