|  |  |
| --- | --- |
| **Largest Number in C++** | |
| #include <iostream>  #include <vector>  #include <algorithm>  using namespace std;  // Custom comparator function for sorting strings in descending order  bool compare(string a, string b) {  string ab = a + b;  string ba = b + a;  return ab > ba; // Compare in descending order  }  string largestNumber(vector<int>& nums) {  // Convert integers to strings  vector<string> arr(nums.size());  for (int i = 0; i < nums.size(); ++i) {  arr[i] = to\_string(nums[i]);  }  // Sort using custom comparator  sort(arr.begin(), arr.end(), compare);  // Construct the result string  if (arr[0] == "0") { // Special case to handle if all nums are zeroes  return "0";  }    string result;  for (const string& s : arr) {  result += s;  }    return result;  }  int main() {  vector<int> nums = {3, 7, 34, 5, 9};  cout << largestNumber(nums) << endl;    return 0;  } | Input: vector<int> nums = {3, 7, 34, 5, 9}; 🔁 Step 1: Convert Integers to Strings  | **Index** | **Integer** | **String** | | --- | --- | --- | | 0 | 3 | "3" | | 1 | 7 | "7" | | 2 | 34 | "34" | | 3 | 5 | "5" | | 4 | 9 | "9" |  📊 Step 2: Custom Sorting (Using compare(a, b) ⇒ a + b > b + a)Sorted Comparisons  | **Pair** | **a + b** | **b + a** | **Result** | | --- | --- | --- | --- | | "9", "5" | "95" | "59" | "9" > "5" | | "9", "34" | "934" | "349" | "9" > "34" | | "5", "3" | "53" | "35" | "5" > "3" | | "7", "3" | "73" | "37" | "7" > "3" | | "34", "3" | "343" | "334" | "34" > "3" |   ➡️ After sorting with custom comparator:   | **Index** | **String** | | --- | --- | | 0 | "9" | | 1 | "7" | | 2 | "5" | | 3 | "34" | | 4 | "3" |  🧩 Step 3: Concatenate Sorted Strings result = "9" + "7" + "5" + "34" + "3" = "975343" ✅ Final Output: 975343 |
| 975343 | |