## Two Sum in C++

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

vector<vector<int>> twoSum(vector<int>
nums, int target) {
    vector<vector<int>> res;
    int n = nums.size();
    sort(nums.begin(), nums.end()); // Sorting
the array

    int left = 0, right = n - 1;
    while (left < right) {
        if (left > 0 && nums[left] == nums[left -
1]) { // Skip duplicates for left pointer
            left++;
            continue;
        }

        int sum = nums[left] + nums[right];

        if (sum == target) {
            res.push_back({nums[left],
nums[right]});
            left++;
            right--;

            // Skip duplicates for both left and
right pointers
            while (left < right && nums[left] ==
nums[left - 1]) left++;
            while (left < right && nums[right] ==
nums[right + 1]) right--;

        } else if (sum > target) {
            right--;
        } else {
            left++;
        }
    }

    return res;
}

int main() {
    vector<int> nums = {2, 2, 4, 3, 1, 6, 6, 7, 5,
9, 1, 8, 9};
    int target = 10;

    vector<vector<int>> res = twoSum(nums,
target);

    // Sorting each pair and then sorting all
pairs lexicographically
    sort(res.begin(), res.end(), [](const
vector<int>& a, const vector<int>& b) {
        return a[0] == b[0] ? a[1] < b[1] : a[0] <
b[0];
    });
```

**Input:**

nums = {2, 2, 4, 3, 1, 6, 6, 7, 5, 9, 1, 8, 9}
target = 10

After sorting:

nums = {1, 1, 2, 2, 3, 4, 5, 6, 6, 7, 8, 9, 9}

**🔍 Step-by-step Table Dry Run:**

| Step | left | right | nums[left] | nums[right] | sum | Action | Result |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 12 | 1 | 9 | 10 | Found a pair, store it | {1, 9} |
| | 1 | 11 | 1 | 9 | 10 | Skip duplicate left | |
| | 2 | 11 | 2 | 9 | 11 | Sum > target, move right-- | |
| 2 | 2 | 10 | 2 | 8 | 10 | Found a pair, store it | {1, 9}, {2, 8} |
| | 3 | 9 | 2 | 7 | 9 | Skip duplicate left, move left++ | |
| 3 | 4 | 9 | 3 | 7 | 10 | Found a pair, store it | {1,9},{2,8}, {3,7} |
| 4 | 5 | 8 | 4 | 6 | 10 | Found a pair, store it | {1,9},{2,8}, {3,7},{4,6} |
| 5 | 6 | 7 | 5 | 6 | 11 | Sum > target, move right-- | |
| 6 | 6 | 6 | 5 | 5 | 10 | Stop (left >= right) | |

**✅ Final Result:**

{ {1, 9}, {2, 8}, {3, 7}, {4, 6} }

```
    // Printing the result
    for (auto& pair : res) {
        for (int val : pair) {
            cout << val << " ";
        }
        cout << endl;
    }

    return 0;
}
```

```
1 9
2 8
3 7
4 6
```