Marks of PCM in C++

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

// Define a class to represent marks
class Marks {
public:
    int phy;
    int chem;
    int math;

    // Constructor
     Marks(int p, int c, int m) {
        phy = p;
        chem = c;
        math = m;
    }

    // Method to compare for sorting
    bool operator<(const Marks& other) const {
        if (phy != other.phy) {
            return phy < other.phy;
        } else if (chem != other.chem) {
            return chem > other.chem; // Sort chem
descending if phy are equal
        } else {
            return math < other.math;
        }
    }
};

// Function to custom sort marks
void customSort(vector<int>& phy, vector<int>&
chem, vector<int>& math) {
    int n = phy.size();
    vector<Marks> arr;

    // Populate the vector of Marks objects
    for (int i = 0; i < n; ++i) {
        arr.emplace_back(phy[i], chem[i], math[i]);
    }

    // Sort using overloaded < operator in Marks class
    sort(arr.begin(), arr.end());

    // Update original arrays with sorted values
    for (int i = 0; i < n; ++i) {
        phy[i] = arr[i].phy;
        chem[i] = arr[i].chem;
        math[i] = arr[i].math;
    }
}

int main() {
    const int N = 5;
    vector<int> phy = {9, 5, 9, 8, 5};
    vector<int> chem = {3, 4, 3, 7, 6};
    vector<int> math = {15, 10, 11, 13, 12};
```

Input Table (Before Sorting)

Index	Phy	Chem	Math
0	9	3	15
1	5	4	10
2	9	3	11
3	8	7	13
4	5	6	12

Sorting Rule Recap

- ✓ Primary: **Phy ascending**
- ✓ Secondary: **Chem descending**
- ✓ Tertiary: **Math ascending**

Output Table (After Sorting)

New Index	Phy	Chem	Math	Reason
0	5	6	12	Smallest phy; chem=6 > chem=4
1	5	4	10	Same phy as above, chem is lower so placed after
2	8	7	13	Next higher phy
3	9	3	11	Same phy as next, but math is smaller so comes first
4	9	3	15	Same phy and chem as above, but math=15 > math=11, so placed after

```
    // Call custom sort function
    customSort(phy, chem, math);

    // Output sorted marks
    for (int i = 0; i < N; ++i) {
        cout << phy[i] << " " << chem[i] << " " << math[i]
<< endl;
    }

    return 0;
}
```

```
5 6 12
5 4 10
8 7 13
9 3 11
9 3 15
```