

Name and marks in different subjects (physics, chemistry and maths) are given for all students. The task is to compute total marks and ranks of all students. And finally display all students sorted by rank.

Rank of student is computed using below rules.

1. If total marks are different, then students with higher marks gets better rank.
2. If total marks are same, then students with higher marks in Maths gets better rank.
3. If total marks are same and marks in Maths are also same, then students with better marks in Physics gets better rank.
4. If all marks (total, Maths, Physics and Chemistry) are same, then any student can be assigned better rank.

We use below structure to store details of students.

```
struct Student
{
    string name; // Given
    int math;    // Marks in math (Given)
    int phy;     // Marks in Physics (Given)
    int che;     // Marks in Chemistry (Given)
    int total;   // Total marks (To be filled)
    int rank;    // Rank of student (To be filled)
};
```

We use `std::sort()` for **Structure Sorting**. In Structure sorting, all the respective properties possessed by the structure object are sorted on the basis of one (or more) property of the object.

```
// C++ program to demonstrate structure sorting in C++
#include <bits/stdc++.h>
using namespace std;

struct Student
{
    string name; // Given
    int math;    // Marks in math (Given)
    int phy;     // Marks in Physics (Given)
    int che;     // Marks in Chemistry (Given)
    int total;   // Total marks (To be filled)
    int rank;    // Rank of student (To be filled)
};
```

```

// Function for comparing two students according
// to given rules
bool compareTwoStudents(Student a, Student b)
{
    // If total marks are not same then
    // returns true for higher total
    if (a.total != b.total )
        return a.total > b.total;

    // If marks in Maths are not same then
    // returns true for higher marks
    if (a.math != b.math)
        return a.math > b.math;

    return (a.phy > b.phy);
}

```

```

// Fills total marks and ranks of all Students
void computeRanks(Student a[], int n)
{
    // To calculate total marks for all Students
    for (int i=0; i<n; i++)
        a[i].total = a[i].math + a[i].phy + a[i].che;

    // Sort structure array using user defined
    // function compareTwoStudents()
    sort(a, a+5, compareTwoStudents);

    // Assigning ranks after sorting
    for (int i=0; i<n; i++)
        a[i].rank = i+1;
}

```

```

// Driver code
int main()
{
    int n = 5;

    // array of structure objects
    Student a[n];

    // Details of Student 1
    a[0].name = "Bryan" ;
    a[0].math = 80 ;
    a[0].phy = 95 ;
    a[0].che = 85 ;

    // Details of Student 2
    a[1].name= "Kevin" ;
    a[1].math= 95 ;
    a[1].phy= 85 ;
    a[1].che= 99 ;
}

```

```

// Details of Student 3
a[2].name = "Nick" ;
a[2].math = 95 ;
a[2].phy = 85 ;
a[2].che = 80 ;

// Details of Student 4
a[3].name = "AJ" ;
a[3].math = 80 ;
a[3].phy = 70 ;
a[3].che = 90 ;

// Details of Student 5
a[4].name = "Howie" ;
a[4].math = 80 ;
a[4].phy = 80 ;
a[4].che = 80 ;

computeRanks(a, n);

//Column names for displaying data
cout << "Rank" <<"t" << "Name" << "t";
cout << "Maths" <<"t" <<"Physics" <<"t"
    << "Chemistry";
cout << "t" << "Totaln";

// Display details of Students
for (int i=0; i<n; i++)
{
    cout << a[i].rank << "t";
    cout << a[i].name << "t";
    cout << a[i].math << "t"
        << a[i].phy << "t"
        << a[i].che << "tt";
    cout << a[i].total <<"t";
    cout <<"n";
}

return 0;
}

```

Output:

Rank	Name	Maths	Physics	Chemistry	Total
1	Kevin	95	85	99	279
2	Nick	95	85	80	260
3	Bryan	80	95	85	260
4	Howie	80	80	80	240
5	AJ	80	70	90	240