

F010A. List of students**P44576_en**

The Boss of the Area of Support to the Decision Making of the FIB has a list with the IDN and the name of certain students of the UPC. On the other hand, he also has a list that enumerates the marks that some students of the UPC have had in certain subjects of certain centers. Both lists are sorted by IDN.

From these two lists, the boss wants to obtain efficiently a list that detaches the marks from each student, and the average of their marks. He has the program started (see `lstlisting` attached) and he has asked you to finish it.

Input

The input consists of two lists, each one sorted by IDN and without repeated lines. The `lstlisting` that defines the data structs and the code that reads the input is already done. Do not modify it!

Output

The output is a list sorted by the IDNs that contains, for each student, his marks and the average of them. The marks must appear in the same order than in the second input list.

Notice that, for each student, you must separate with five white spaces the information of his marks, and then, you must print an empty line. Notice also that if an identifier appears in an input list but it does not appear in the other one, it has not to appear in the output list.

Sample input 1

```
5
1312 Charlotte
5561 Natalie
7790 Laura
8876 Smith
9386 Martin

7
1312 P1 FIB 7
1312 AL FIB 8
2020 IC FIB 4.8
5561 P1 FIB 0
5561 I1 FME 6.5
7790 P1 FIB 10
9386 I1 FME 10
```

Sample output 1

```
1312 Charlotte
    P1 FIB 7.00
    AL FIB 8.00
    average: 7.50

5561 Natalie
    P1 FIB 0.00
    I1 FME 6.50
    average: 3.25

7790 Laura
    P1 FIB 10.00
    average: 10.00

9386 Martin
    I1 FME 10.00
    average: 10.00
```

Sample input 2

```
1
123456789 Mary
```

```
1
987654321 P1 FIB 5
```

Sample output 2

```

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

struct Student {
    int idn;
    string name;
};

struct Mark {
    int idn;
    string subj;      // subject
    string sch;      // school
    double mark;
};

void read_students( vector<Student>& students) {
    int m;
    cin >> m;
    students = vector<Student>(m);
    for (int i = 0; i < m; ++i) cin >> students[i].idn >> students[i].name;
}

void read_marks( vector<Mark>& marks) {
    int n;
    cin >> n;
    marks = vector<Mark>(n);
    for (int i = 0; i < n; ++i) {
        cin >> marks[i].idn >> marks[i].subj >> marks[i].sch >> marks[i].mark;
    }
}

int main() {
    cout . setf( ios :: fixed );      //Set the the format of doubles
    cout . precision (2);           // to write the marks

    vector<Student> students;
    vector<Mark> marks;
    reads_students (students );
    reads_marks(marks);

    ...
}

```

Problem information

Author: Professorat de P1

Translator: Carlos Molina

Generation: 2026-01-25T11:14:46.898Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>