

## → NESTING

23

→ let's suppose we have to take input like this :

3 // Test cases

0 6 4 // First name, Last name,  
no. of subjects.

1 2 3 4 // marks in each subjects  
respectively.

d f 3

2 3 4

c d 2

1 2

→ How a pair is sorted in as a map's key :

→ First element of each pair is checked whichever is smaller comes first if two of them are equal. ex:

2 3 and 2 4

So, here 1<sup>st</sup> element of both the pairs is same, so comparison comes to second elements of pair and as they are different, key of map get sorted according to them. like this:

2 3

2 4

Same process follows for vector and set if they are used as a key in map.

→ Defining a map with pair as a key and vector as a value.

```
MAP < PAIR < STRING, STRING >, VECTOR < INT >> M;
```

→ No. of inputs we want to take in map

```
INT N;
```

```
CIN >> N;
```

→ Taking input in map

```
FOR (INT i = 0; i < n; i++)
```

```
{
```

```
    STRING FN, LN;
```



INT CT;

→ Taking first name, last name and no. of subject as an input.

CTN >> FN >> LN >> CT;

→ Taking marks of each subject

```
FOR (INT j=0; j < CT; j++)
{
```

INT x;

CTN >> x;

M[{FN, LN}].PB(x);

```
}
```

→ Printing out nested map (We are ~~taking~~ passing every data as type as a reference so that no copy of any of them is created as making copies will increase our time complexity.)

FOR (AUTO &PR : M)

{

→ Extracting pair

AUTO &FULL-NAME = PR.F;

→ Extracting vector

AUTO &LIST = PR.S;

→ Extracting both values of a pair

COUT << FULL-NAME.F << " " << FULL-NAME.S << "\n";

→ Extracting size of the vector.

COUT << LIST.SIZE() << "\n";

```

-> Printing elements of the vector
FOR (AUTO & ELEMENT : L33)
{
    cout << ELEMENT << " ";
}
cout << "\n";
}

```

-> OUTPUT:

a b

4

1 2 3 4

c d

2

1 2

d f

3

2 3 4