# Organizing Containers of Balls

🔒 locked

by ma5termind

Problem Submissions Leaderboard Discussions **Editorial**

### Editorial by Shafaet

As the operations only allow us to swap balls between two containers, each container will contain a fixed number of balls all the time. Suppose the $i^{th}$ container always contains $cap_i$ number of balls. Each $cap_i$ can be calculated by finding the sum of row $i$ in matrix $M$.

Let $ball_j$ denotes the number of balls of color $j$. Each $ball_j$ can be calculated by finding the sum of column $j$ in matrix $M$.

Let's think about container $i$ and ball of type $j$. As container $i$ will always contain $cap_i$ number of balls, ball of type $j$ can fit in container $i$ only of $ball_j = cap_i$.

Now for each $ball_j$ you have to find a suitable container $i$ such that $ball_j = cap_i$. If you can find such container for each balls, it's possible to obtain the desired configuration. It can be done by simply sorting the array $ball$ and $cap$. After sorting check if the arrays are exactly same or not. If the arrays are same, it's possible to obtain the desired configuration, otherwise it's impossible.

## Python

```
tests = int (input ())
for test in range (tests):
    n = int (input ())
    sr = [0] * n
    sc = [0] * n
    for i in range (n):
        row = map(int, raw_input().split())
        for j in range (n):
            sr[i] += row[j]
            sc[j] += row[j]
    print sr
    print sc
    print ("Possible" if sorted (sr) == sorted (sc) else "Impossible")
```

### Set by ma5termind

Problem Setter's code :

## CPP

```
#include <bits/stdc++.h>
// #include "testlib.h"
using namespace std ;

#define ft first
#define sd second
#define pb push_back
#define all(x) x.begin(),x.end()

#define ll long long int
#define vi vector<int>
#define vii vector<pair<int,int> >
#define pii pair<int,int>
#define plii pair<pair<ll, int>, int>
#define piii pair<pii, int>
#define viii vector<pair<pii, int> >
#define vl vector<ll>
#define vll vector<pair<ll,ll> >
```

## Statistics

Difficulty: **Medium**

Time Complexity: O(T*(N^2

Required Knowledge: **Sorting, Obs**

Publish Date: **Jan 23 201**

```
#define scll1(x) scanf("%lld",&x)
#define scll2(x,y) scanf("%lld%lld",&x,&y)
#define scll3(x,y,z) scanf("%lld%lld%lld",&x,&y,&z)

#define pr1(x) printf("%d\n",x)
#define pr2(x,y) printf("%d %d\n",x,y)
#define pr3(x,y,z) printf("%d %d %d\n",x,y,z)

#define prll1(x) printf("%lld\n",x)
#define prll2(x,y) printf("%lld %lld\n",x,y)
#define prll3(x,y,z) printf("%lld %lld %lld\n",x,y,z)

#define pr_vec(v) for(int i=0;i<v.size();i++) cout << v[i] << " " ;

#define f_in(st) freopen(st,"r",stdin)
#define f_out(st) freopen(st,"w",stdout)

#define fr(i, a, b) for(i=a; i<=b; i++)
#define fb(i, a, b) for(i=a; i>=b; i--)
#define ASST(x, l, r) assert( x <= r && x >= l )

#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

const int mod = 1e9 + 7;

int ADD(int a, int b, int m = mod) {
    int s = a;
    s += b;
    if( s >= m )
      s -= m;
    return s;
}

int MUL(int a, int b, int m = mod) {
    return (1LL * a * b % m);
}

int power(int a, int b, int m = mod) {
    int res = 1;
    while( b ) {
        if( b & 1 ) {
            res = 1LL * res * a % m;
        }
        a = 1LL * a * a % m;
        b /= 2;
    }
    return res;
}

ll nC2(ll x) {
    return ( x * ( x - 1 ) / 2 );
}

const int maxn = 1e5 + 5;

int main() {
        int t; cin >> t;
    assert(t <= 10);
        while( t-- ) {
                int n; cin >> n;
        assert(n <= 100);
                ll row[n], col[n];
                int i, j;
                fr(i, 0, n-1) row[i] = col[i] = 0;
                fr(i, 0, n-1) {
                        fr(j, 0, n-1) {
                                int x; cin >> x;
                assert(x <= 1000000000);
                                row[i] += x; col[j] += x;
                        }
                }
                sort(row, row+n);
                sort(col, col+n);
                bool ok = true;
                fr(i, 0, n-1) {
            assert(row[i] >= 0 && col[i] >= 0);
```

### Tested by dansagunov

Problem Tester's code :

## Python3

```
tests = int (input ())
for test in range (tests):
    n = int (input ())
    sr = [0] * n
    sc = [0] * n
    for i in range (n):
        row = list (map (int, input ().split (' ')))
        for j in range (n):
            sr[i] += row[j]
            sc[j] += row[j]
    print ("Possible" if sorted (sr) == sorted (sc) else "Impossible")
```