

[Sắp xếp - Tìm Kiếm]. Bài 11. Trộn 2 dãy

Để sắp xếp tăng dần một mảng A gồm n phần tử a_1, a_2, \dots, a_n , thuật toán sắp xếp trộn (MergeSort) áp dụng chia đôi mảng A thành hai mảng B và C, sắp xếp B, C và sau đó trộn B và C cho ra mảng A tăng dần. Ví dụ minh họa phương pháp trộn: - Mảng B gồm 4 phần tử b_1, b_2, b_3, b_4 đã sắp tăng dần: 1 2 4 6 - Mảng C gồm 4 phần tử c_1, c_2, c_3, c_4 đã sắp tăng dần: 3 5 8 9 Nếu trộn hai mảng trên theo dãy thứ tự trộn $b_1, b_2, c_1, b_3, c_2, b_4, c_3, c_4$ thì có được mảng sắp là 1 2 3 4 5 6 8 9. Cho một mảng B gồm n phần tử và mảng C gồm m phần tử. Hãy in ra dãy thứ tự trộn sao cho nếu áp dụng dãy thứ tự trộn trên thì mảng kết quả được sắp xếp tăng dần.

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

using ll = long long;

int main(){
    int n, m; cin >> n >> m;
    int b[n], c[m];
    for(int i = 0; i < n; i++) cin >> b[i];
    for(int i = 0; i < m; i++) cin >> c[i];
    int i = 0, j = 0;
    while(i < n && j < m){
        if(b[i] <= c[j]){
            cout << "b" << i + 1 << " ";
            ++i;
        }
        else{
            cout << "c" << j + 1 << " ";
            ++j;
        }
    }
    while(i < n){
        cout << "b" << i + 1 << " ";
        ++i;
    }
    while(j < m){
        cout << "c" << j + 1 << " ";
        ++j;
    }
}
```

Input Format

- Dòng đầu tiên là hai số nguyên n, m cách nhau một khoảng trắng là số phần tử của mảng B và mảng C.
- Dòng thứ 2 gồm n số nguyên b_1, b_2, \dots, b_n , mỗi số cách nhau một khoảng trắng.
- Dòng thứ 3 gồm m số nguyên c_1, c_2, \dots, c_m , mỗi số cách nhau một khoảng trắng.

Constraints

$1 \leq n, m \leq 10^6$; $0 \leq b[i], c[i] \leq 10^6$

Output Format

In ra dãy là thứ tự trộn, trong trường hợp trong 2 mảng b và c có phần tử có cùng giá trị t in ra phần tử ở mảng b trước.

Sample Input 0

```
5 5
4760 9724 9798 20124 25974
3397 9166 13054 18273 30455
```

Sample Output 0

```
c1 b1 c2 b2 b3 c3 c4 b4 b5 c5
```