

### [Set Map]. Bài 1. Phần tử phân biệt

Cho dãy số  $A[]$  gồm có  $N$  phần tử, bạn hãy đếm xem trong mảng có bao nhiêu phần tử phân biệt? Chú ý giải bài này với 3 cách : Sử dụng set, map, sắp xếp.

#### Input Format

Dòng đầu tiên là số nguyên  $N$ . Dòng tiếp theo gồm  $N$  số nguyên  $A[i]$

#### Constraints

$$1 \leq N \leq 10^5; -10^9 \leq A[i] \leq 10^9$$

#### Output Format

In ra số lượng phần tử khác nhau trong mảng.

#### Sample Input 0

```
5
2 1 2 1 3
```

#### Sample Output 0

```
3
```

### [Set Map]. Bài 2. Truy vấn phần tử trong mảng.

Cho dãy số  $A[]$  gồm có  $N$  phần tử, bạn hãy trả lời các truy vấn để xác định xem phần tử  $X$  nào đó có nằm trong mảng hay không? Bạn hãy thử giải bài này bằng 3 cách : Set, Map, Binary Search.

#### Input Format

- Dòng đầu tiên là số nguyên  $N$ .
- Dòng thứ 2 gồm  $N$  số nguyên  $A[i]$
- Dòng thứ 3 là số lượng truy vấn  $Q$ .
- $Q$  dòng tiếp theo mỗi dòng là một số nguyên  $X$ .

#### Constraints

$$1 \leq N \leq 10^5; 0 \leq A[i], X \leq 10^9; 1 \leq Q \leq 1000;$$

#### Output Format

Đối với mỗi truy vấn in ra YES nếu phần tử  $X$  xuất hiện trong mảng, ngược lại in ra NO.

#### Sample Input 0

```
5
1 2 1 3 4
2
1
6
```

#### Sample Output 0

```
YES
NO
```

#### [Set Map]. Bài 3. Phần tử riêng biệt.

Cho dãy số  $A[]$  gồm có  $N$  phần tử, bạn hãy liệt kê các giá trị xuất hiện trong dãy theo thứ tự xuất hiện, mỗi giá trị chỉ liệt kê một lần.

Gợi ý : Dùng set để lưu các phần tử đã được in ra trước đó, gặp phần tử  $a[i]$  thì kiểm tra nhanh trong set có giá trị  $a[i]$  chưa để in ra

#### Input Format

Dòng đầu tiên là số nguyên  $N$ . Dòng thứ 2 gồm  $N$  số nguyên  $A[i]$

#### Constraints

$1 \leq N \leq 10^5$ ;  $0 \leq A[i], X \leq 10^9$ ;

#### Output Format

In ra các giá trị xuất hiện trong mảng theo thứ tự xuất hiện.

#### Sample Input 0

```
8
1 2 3 2 1 3 4 8
```

#### Sample Output 0

```
1 2 3 4 8
```

#### [Set Map]. Bài 4. Truy vấn trên mảng

Cho dãy số  $A[]$  gồm có  $N$  phần tử, có 3 thao tác như sau :

- Thao tác 1 : Thêm 1 phần tử  $X$  vào mảng.
- Thao tác 2 : Xóa 1 phần tử  $X$  khỏi mảng. Trong trường hợp phần tử  $X$  không xuất hiện trong mảng, sẽ không thực hiện xóa, nếu trong mảng có nhiều phần tử  $X$  thì chỉ xóa đi 1 phần tử  $X$  trong mảng.
- Thao tác 3 : Truy vấn xem phần tử  $X$  có xuất hiện trong mảng hay không?

Ban đầu mảng A có N phần tử, với các truy vấn phần tử X trong mảng, in ra YES nếu X xuất hiện trong mảng, ngược lại in ra NO.

### Input Format

- Dòng đầu tiên là số nguyên N.
- Dòng thứ 2 gồm N số nguyên A[i].
- Dòng thứ 3 là số lượng thao tác Q.
- Q dòng tiếp theo mỗi dòng là thông tin của một thao tác, gồm 2 số, số đầu tiên là loại thao tác, số tiếp theo là phần tử X trong thao tác.

### Constraints

$1 \leq N \leq 10^4$ ;  $0 \leq A[i], X \leq 10^9$ ;  $1 \leq Q \leq 1000$ ;  $0 \leq X \leq 10^9$ ;

### Output Format

Đối với mỗi thao tác loại 3, in ra YES nếu X xuất hiện trong mảng, ngược lại in ra NO.

### Sample Input 0

```
6
1 2 3 4 5 6
4
1 1
2 6
3 1
3 6
```

### Sample Output 0

```
YES
NO
```

### Explanation 0

Ban đầu mảng có 6 phần tử 1, 2, 3, 4, 5, 6. Sau thao tác đầu tiên mảng có thêm 1 phần tử 1 và trở thành mảng (1, 1, 2, 3, 4, 5, 6). Sau thao tác thứ 2, mảng xóa đi 1 phần tử 6 và trở thành mảng (1, 1, 2, 3, 4, 5). Thao tác thứ 3 truy vấn phần tử 1, câu trả lời là YES vì 1 xuất hiện trong mảng. Thao tác thứ 4 truy vấn phần tử 6, câu trả lời là NO vì 6 không xuất hiện trong mảng.

### [Set Map]. Bài 16. Tìm kiếm sinh viên.

Ở trường đại học xyz, mỗi sinh viên sẽ có một mã sinh viên riêng. Mã sinh viên là một chuỗi ký tự không quá 8 ký tự. Bạn được yêu cầu xây dựng chương trình có thể kiểm tra một mã sinh viên nào đó có thuộc về sinh viên nào?

```
//Gợi ý dùng map<string, string> mp;  
//Trong đó key thì lưu mã sinh viên và value thì lưu tên
```

### Input Format

- Dòng đầu tiên là số lượng sinh viên N.
- 2N dòng tiếp theo là các dòng tiếp theo mô tả thông tin của sinh viên trên 2 dòng, dòng đầu là mã sinh viên, dòng 2 là tên sinh viên.
- Dòng tiếp theo là số truy vấn Q.
- Q dòng tiếp theo, mỗi dòng là một mã sinh viên cần tìm kiếm, nếu mã sinh viên này thuộc về một bạn sinh viên thì in ra tên của sinh viên đó trên 1 dòng, ngược lại in ra "NOT FOUND" trên 1 dòng.

### Constraints

- $1 \leq N \leq 10^4$ ;  $1 \leq Q \leq 1000$
- Mã sinh viên là xâu kí tự không quá 8 kí tự
- Tên sinh viên là một xâu có không quá 30 kí tự

### Output Format

In ra kết quả của từng truy vấn, mỗi truy vấn trên 1 dòng.

### Sample Input 0

```
3  
SV001  
Hoang Van Nam  
SV002  
Ngo Gia Long  
SV005  
Phuong The Ngoc  
2  
SV005  
SV003
```

### Sample Output 0

```
Phuong The Ngoc  
NOT FOUND
```

### [Set Map]. Bài 17. Truy vấn kí tự

Cho một xâu kí tự S chỉ bao gồm các kí tự từ a tới z. Bạn được yêu cầu thực hiện các nhiệm vụ sau bằng CTDL phù hợp. - 1. Tìm kí tự có tần suất xuất hiện nhiều nhất và có thứ tự từ điển nhỏ nhất.

- 
- 2. Tìm kí tự có tần suất xuất hiện nhỏ nhất và có thứ tự từ điển lớn nhất.
- 
- 3. Tìm số lượng kí tự khác nhau trong xâu.

### Input Format

Xâu kí tự S.

### Constraints

$1 \leq \text{len}(S) \leq 10^6$ ;

### Output Format

- Dòng đầu tiên in ra kí tự có tần suất lớn nhất, nếu có nhiều kí tự có cùng tần suất in ra kí tự có thứ tự từ điển nhỏ nhất.
- Dòng thứ 2 in ra kí tự có tần suất nhỏ nhất, nếu có nhiều kí tự có cùng tần suất thì in ra kí tự có thứ tự từ điển lớn nhất.
- Dòng 3 in ra số lượng kí tự khác nhau trong xâu.

### Sample Input 0

```
mghzbzhcrucvwjssatuhlwcraawtbtmyqjdjksyskvhrqubvtoimrjwnanfqgenshrjtf
```

### Sample Output 0

```
h 5
o 1
24
```

### [Set Map]. Bài 18. Số lượng từ khác nhau trong xâu

Cho một xâu kí tự S chỉ bao gồm các kí tự là chữ cái và dấu cách. Hãy đếm số lượng từ khác nhau trong xâu S và in ra từ có thứ tự từ điển nhỏ nhất, lớn nhất trong xâu S.

Gợi ý : Dùng set;

```
//Cú pháp để đọc hết mọi từ trong luồng vào
```

```
string s;
while(cin >> s){
    //Xử lý s
}
```

### Input Format

Xâu kí tự S.

### Constraints

$$1 \leq \text{len}(S) \leq 10^6;$$

### Output Format

Dòng đầu tiên in ra số lượng từ khác nhau trong xâu. Dòng thứ 2 in ra từ có thứ tự từ điển nhỏ nhất và lớn nhất trong xâu.

### Sample Input 0

```
ngo ngoc an nguyen an ngoc ngoc
```

### Sample Output 0

```
4
an nguyen
```

### [Set Map]. Bài 8. Lớn nhất, nhỏ nhất

Cho mảng  $A[]$  gồm  $n$  phần tử, có 4 loại thao tác

- Thao tác 1 : Thêm phần tử  $X$  vào mảng
- Thao tác 2 : Xóa mọi giá trị  $X$  khỏi mảng nếu  $X$  tồn tại trong mảng
- Thao tác 3 : Tìm phần tử nhỏ nhất trong mảng
- Thao tác 4 : Tìm phần tử lớn nhất trong mảng

Đối với thao tác 3 và 4 bạn hãy in ra phần tử lớn nhất, nhỏ nhất trong mảng  $A[]$

### Input Format

- Dòng 1 gồm  $N$  : Số lượng phần tử ban đầu trong mảng  $A[]$
- Dòng thứ 2 gồm  $N$  phần tử ban đầu của mảng  $A[]$
- Dòng 3 gồm số lượng thao tác  $Q$
- $Q$  dòng tiếp theo mỗi dòng mô tả 1 thao tác, trong đó nếu là thao tác 1 và 2 thì có thêm phần tử  $X$ .

### Constraints

- $1 \leq N \leq 10^5$
- $1 \leq Q \leq 10000$
- $0 \leq A[i], X \leq 10^6$

## Output Format

In ra kết quả với thao tác 3 và 4

### Sample Input 0

```
10
24 31 49 65 57 77 7 81 77 79
10
2 60
2 7
3
3
2 45
1 16
4
3
1 6
1 8
```

### Sample Output 0

```
24
24
81
16
```

## [Set Map]. Bài 9. Union

Cho mảng  $A[]$  và  $B[]$  lần lượt gồm  $N$  và  $M$  số nguyên, nhiệm vụ của bạn là tìm hợp của 2 mảng này và in theo thứ tự từ lớn về nhỏ.

Gợi ý : Để tìm hợp của 2 mảng thì chỉ cần dùng 1 set và lưu mọi phần tử của 2 mảng, set sẽ tự loại bỏ các phần tử trùng nhau trong cả 2 mảng.

Cú pháp duyệt ngược của set như sau :

```
for(auto it = se.rbegin(); it != se.rend(); ++it){
    cout << *it << ' ';
}
```

## Input Format

- Dòng 1 gồm  $N$  và  $M$
- Dòng 2 gồm  $N$  số trong mảng  $A[]$
- Dòng 3 gồm  $M$  số trong mảng  $B[]$

## Constraints

- $1 \leq N, M \leq 10^5$

- $-10^9 \leq A[i], B[i] \leq 10^9$

### Output Format

In ra tập hợp của 2 mảng theo thứ tự giảm dần

### Sample Input 0

```
9 6
7 9 2 0 0 5 6 4 7
9 7 3 4 5 7
```

### Sample Output 0

```
9 7 6 5 4 3 2 0
```

## [Set Map]. Bài 10. Intersection

Cho mảng  $A[]$  và  $B[]$  lần lượt gồm  $N$  và  $M$  số nguyên, nhiệm vụ của bạn là tìm giao của 2 mảng này và in theo thứ tự xuất hiện trong mảng  $A[]$ .

Gợi ý :

Cách 1 : Dùng 2 set để lưu 2 mảng  $A, B$  sau đó duyệt 1 trong 2 mảng và tìm kiếm sự xuất hiện của phần tử đang duyệt trong mảng còn lại.

Cách 2 : Dùng 1 map để đánh dấu, duyệt qua các phần tử trong mảng  $A[]$  và cho map của từng phần tử trong mảng  $A[] = 1$ , duyệt qua mảng  $B[]$  thì kiểm tra nếu map của phần tử trong mảng  $B$  đã bằng 1 thì chuyển thành 2. Cuối cùng duyệt map và in ra những phần tử được đánh dấu là 2.

### Input Format

- Dòng 1 gồm  $N$  và  $M$
- Dòng 2 gồm  $N$  số trong mảng  $A[]$
- Dòng 3 gồm  $M$  số trong mảng  $B[]$

### Constraints

- $1 \leq N, M \leq 10^5$
- $-10^9 \leq A[i], B[i] \leq 10^9$

### Output Format

In ra tập hợp của 2 mảng theo thứ tự xuất hiện trong mảng  $A[]$

### Sample Input 0



```
9 8
7 5 0 0 1 6 2 7 3
6 4 9 4 2 8 3 5
```

#### Sample Output 0

```
5 6 2 3
```

#### [Set Map]. Bài 11. Difference

Cho mảng  $A[]$  và  $B[]$  lần lượt gồm  $N$  và  $M$  số nguyên, nhiệm vụ của bạn là tìm những phần tử xuất hiện trong mảng thứ nhất nhưng không xuất hiện trong mảng thứ 2, liệt kê theo thứ tự từ bé đến lớn.

Gợi ý : Dùng 2 set để lưu mảng A, B. Duyệt các phần tử trong set 1 và tìm kiếm nếu phần tử ko nằm trong set 2 thì in ra.

#### Input Format

- Dòng 1 gồm  $N$  và  $M$
- Dòng 2 gồm  $N$  số trong mảng  $A[]$
- Dòng 3 gồm  $M$  số trong mảng  $B[]$

#### Constraints

- $1 \leq N, M \leq 10^5$
- $-10^9 \leq A[i], B[i] \leq 10^9$

#### Output Format

In ra đáp án của bài toán

#### Sample Input 0

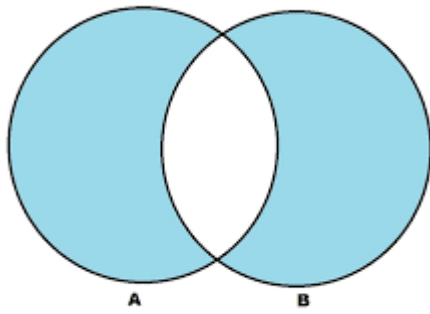
```
6 6
1 9 3 8 0 7
3 1 7 3 7 5
```

#### Sample Output 0

```
0 8 9
```

#### [Set Map]. Bài 12. Symmetric difference

Cho mảng  $A[]$  và  $B[]$  lần lượt gồm  $N$  và  $M$  số nguyên, nhiệm vụ của bạn là tìm những phần tử xuất hiện ở 1 trong 2 mảng nhưng không được xuất hiện ở cả 2 mảng.



### Input Format

- Dòng 1 gồm N và M
- Dòng 2 gồm N số trong mảng A[]
- Dòng 3 gồm M số trong mảng B[]

### Constraints

- $1 \leq N, M \leq 10^5$
- $-10^9 \leq A[i], B[i] \leq 10^9$

### Output Format

In ra đáp án của bài toán theo thứ tự từ bé đến lớn

### Sample Input 0

```
9 8
9 0 2 3 9 7 8 1 4
1 0 2 1 1 2 0 5
```

### Sample Output 0

```
3 4 5 7 8 9
```

### [Set Map]. Bài 13. Loại trừ

Cho mảng A[] và B[] có N và M phần tử là các số nguyên, hãy tìm các phần tử xuất hiện trong mảng 1 mà không xuất hiện trong mảng 2.

### Input Format

Dòng đầu tiên là N và M; Dòng thứ 2 là N số của mảng A[]; Dòng thứ 3 là M số của mảng B[];

### Constraints

$1 \leq N, M \leq 10^6$ ;  $-10^9 \leq A[i], B[i] \leq 10^9$ ;

### Output Format

In ra các số xuất hiện trong mảng 1 mà không xuất hiện trong mảng 2 theo thứ tự từ bé đến lớn, mỗi giá trị thỏa mãn chỉ liệt kê 1 lần.

### Sample Input 0

```
3 4
1 2 3
1 2 4 5
```

### Sample Output 0

```
3
```

### [Set Map]. Bài 14. Suffix And Query

Cho mảng  $A[]$  gồm  $N$  phần tử. Bạn được yêu cầu trả lời cho các truy vấn, mỗi truy vấn yêu cầu bạn đếm số các giá trị khác nhau từ chỉ số  $L$  tới chỉ số  $N - 1$ .

Gợi ý : Gọi  $F[i]$  là số lượng các giá trị khác nhau tính từ chỉ số  $i$  tới chỉ số  $n - 1$  trong mảng, xây dựng trước mảng  $F[i]$ , sau đó mỗi truy vấn chỉ cần truy cập vào  $F[l]$  là có thể tìm được số lượng các giá trị khác nhau từ  $l$  tới  $n - 1$ . Cách xây dựng mảng  $F[]$  là duyệt từ cuối về và dùng 1 set để lưu các giá trị khác nhau tính từ cuối, gán  $F[i] = \text{set.size}()$

### Input Format

- Dòng đầu tiên là  $N$  - số lượng phần tử của mảng
- Dòng thứ 2 gồm  $N$  phần tử  $A[i]$  của mảng
- Dòng thứ 3 là số truy vấn  $Q$
- $Q$  dòng tiếp theo mỗi dòng là một số  $L$ .

### Constraints

- $1 \leq N \leq 1000000$
- $1 \leq A[i] \leq 10^9$
- $1 \leq Q \leq 10000$
- $0 \leq L \leq N - 1$

### Output Format

Với mỗi truy vấn in kết quả trên 1 dòng.

### Sample Input 0

```
9
3 3 3 4 2 0 3 1 2
4
8
6
1
7
```

#### Sample Output 0

```
1
3
5
2
```

#### [Set Map]. Bài 15. Set vs Lower\_bound, Upper\_bound

Cho 1 mảng  $A[]$  gồm  $N$  số nguyên, 28tech cung cấp cho bạn các thao tác sau :

- 1 x : Thêm phần tử  $x$  vào mảng  $A[]$
- 2 x : Xóa phần tử  $x$  khỏi mảng  $A[]$  nếu  $x$  tồn tại trong mảng, và nếu  $x$  xuất hiện nhiều lần bạn chỉ xóa đi 1 phần tử duy nhất.
- 3 x : Chỉ ra giá trị đầu tiên của phần tử  $\geq x$  trong  $A[]$  khi mảng  $A[]$  được sắp xếp tăng dần, nếu không tồn tại in ra -1
- 4 x : In ra giá trị của phần tử lớn nhất  $\leq x$  trong mảng  $A[]$ , nếu không tồn tại phần tử này thì in ra -1.

#### Input Format

- Dòng 1 là  $N$
- Dòng 2 là  $N$  phần tử  $A[i]$
- Dòng 3 là số lượng thao tác  $T$
- $T$  dòng tiếp theo mô tả thao tác.

#### Constraints

- $1 \leq N \leq 1000$
- $1 \leq A[i], X \leq 1000$ ;
- $1 \leq T \leq 1000$

#### Output Format

Đối với thao tác 3 và 4, in ra kết quả tương ứng

#### Sample Input 0

```
9
6 4 9 3 8 6 8 1 6
9
4 3
4 6
3 6
3 4
1 6
2 6
2 7
4 4
1 4
```

#### Sample Output 0

```
3
6
6
4
4
```

#### [Set Map]. Bài 27. Thi đấu

Cho thông tin các trận đấu của giải bóng đá Sunday League, nhiệm vụ của bạn là đối với mỗi đội bóng hãy liệt kê các đối thủ đã từng chạm trán. Danh sách các đội bóng được liệt kê theo thứ tự từ điển tăng dần và danh sách đối thủ của từng đội bóng cũng được liệt kê tăng dần theo thứ tự từ điển.

Gợi ý : Dùng `map(string, vector(string)) mp`; mỗi đội bóng sẽ dùng 1 vector để lưu lại danh sách các đối thủ trong các trận bóng, sort danh sách đội bóng trước khi in ra kết quả.

#### Input Format

- Dòng 1 là  $N$  : số trận bóng diễn ra
- $N$  dòng tiếp theo mỗi dòng gồm thông tin của 1 trận đấu theo cú pháp  $X - Y$ , trong đó đội  $X$  thi đấu với đội  $Y$

#### Constraints

- $1 \leq N \leq 1000$

#### Output Format

- In ra kết quả của bài toán theo mẫu

#### Sample Input 0

```
13
```

Arsenal - Lyon  
Fullham - Liverpool  
Fullham - Chelsea  
Barcelona - Chelsea  
Barcelona - Fullham  
Lyon - Barcelona  
Chelsea - AC Milan  
PSG - Manchester City  
Arsenal - Fullham  
Fullham - Arsenal  
Real Madrid - Barcelona  
Arsenal - Manchester City  
Manchester United - Liverpool

### Sample Output 0

AC Milan : Chelsea  
Arsenal : Fullham, Fullham, Lyon, Manchester City  
Barcelona : Chelsea, Fullham, Lyon, Real Madrid  
Chelsea : AC Milan, Barcelona, Fullham  
Fullham : Arsenal, Arsenal, Barcelona, Chelsea, Liverpool  
Liverpool : Fullham, Manchester United  
Lyon : Arsenal, Barcelona  
Manchester City : Arsenal, PSG  
Manchester United : Liverpool  
PSG : Manchester City  
Real Madrid : Barcelona

### [Set Map]. Bài 37. Điểm trung bình

Cho thông tin điểm thi của các môn học của các sinh viên, bạn hãy tính điểm trung bình của sinh viên đó và in ra màn hình. Điểm trung bình được tính bằng cách lấy tổng hệ số điểm và số tín chỉ chia cho tổng số tín chỉ. Ví dụ sinh viên X học môn A 2 tín chỉ và có điểm là 5, môn B 3 tín chỉ và có điểm là 4 thì điểm trung bình được tính =  $(2 * 5 + 3 * 4) / (2 + 3)$ .

### Input Format

- Gồm nhiều dòng, mỗi dòng gồm 3 thông tin : Tên sinh viên (có 1 từ), số tín chỉ và điểm số tương ứng. Dữ liệu đảm bảo không có 2 sinh viên có cùng tên.

### Constraints

- Điểm là số nguyên từ 0 tới 10, số tín chỉ là số nguyên dương

### Output Format

- In ra danh sách sinh viên theo thứ tự từ điển giảm dần và điểm trung bình lấy 2 số sau dấu phẩy.

### Sample Input 0

Lan 4 8  
Hanh 4 5  
Lan 6 10  
Phong 4 10

Nam 4 9  
Phong 6 5  
Lan 2 4  
Phuong 6 1  
Nhung 2 7  
Nhung 6 9  
Nhung 5 8  
Nhung 4 10  
Lan 3 1

#### **Sample Output 0**

Phuong : 1.00  
Phong : 7.00  
Nhung : 8.71  
Nam : 9.00  
Lan : 6.87  
Hanh : 5.00

#### **Sample Input 1**

Nhung 5 1  
Phong 2 6  
Ngoc 5 8  
Phuong 3 10  
Ngoc 3 4  
Nhung 2 10  
Ngoc 3 3  
Nam 5 1  
Nhung 3 5  
Ngoc 4 7  
Lan 3 5  
Phong 5 2  
Phuong 6 3  
Nhung 4 1  
Nhung 4 5

#### **Sample Output 1**

Phuong : 5.33  
Phong : 3.14  
Nhung : 3.56  
Ngoc : 5.93  
Nam : 1.00  
Lan : 5.00