# [Mảng 1 chiều cơ bản]. Bài 1. Số nhỏ nhất

Cho mảng số nguyên A[] gồm N phần tử, hãy đếm xem trong mảng của bạn có bao nhiêu số có cùng giá trị nhỏ nhất. Ví dụ mảng A = {1, 2, 1, 3, 5} thì số nhỏ nhất trong mảng là 1 xuất hiện 2 lần.

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; -10^3<=A[i]<=10^3;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

5

-854 600 222 472 207

**Sample Output 0**

1

# [Mảng 1 chiều cơ bản]. Bài 2.1 Cặp số 1

Cho mảng số nguyên A[] gồm N phần tử, hãy đếm xem trong mảng A[] tồn tại bao nhiêu cặp số A[i], A[j] với i khác j sao cho tổng của 2 phần tử này bằng số K cho trước.

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng; Dòng thứ 3 là số nguyên K

**Constraints**

1<=N<=1000; 0<=A[i], X<=10^3;

**Output Format**

In ra số lượng cặp thỏa mãn

**Sample Input 0**

5

1 2 3 1 2

3

**Sample Output 0**

4

# [Mảng 1 chiều cơ bản]. Bài 2.2 Cặp số 2

Cho mảng số nguyên A[] gồm N phần tử, hãy tìm độ chênh lệch nhỏ nhất giữa 2 phần tử trong mảng.

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng;

**Constraints**

1<=N<=1000; 0<=A[i], X<=10^3;

**Output Format**

In ra độ chênh lệch nhỏ nhất giữa 2 phần tử bất kì trong mảng

**Sample Input 0**

8

69 96 93 27 84 32 78 56

**Sample Output 0**

3

# [Mảng 1 chiều cơ bản]. Bài 3. Liệt kê các giá trị khác nhau

Cho mảng số nguyên A[] gồm N phần tử, hãy liệt kê các giá trị khác nhau trong mảng theo thứ tự xuất hiện, mỗi giá trị chỉ liệt kê 1 lần. Ở thời điểm hiện tại các bạn có thể for trâu để giải bài này, sau này sẽ dùng cách tối ưu hơn.

Gợi ý : Duyệt qua từng chỉ số i của mảng, nếu kiểm tra được đây là giá trị đầu tiên mà bạn gặp thì sẽ in ra giá trị này, tức là trước đó trong đoạn từ 0 tới chỉ số i - 1 ko có giá trị nào giống A[i], ví dụ mảng A = {1, 1, 3, 1, 5, 1, 8}, khi bạn xét số 5 bạn sẽ ra nó vì trước nó ko có giá trị nào giống nó cả. Để mà kiểm tra có in ra giá trị A[i] hay không bạn làm 1 vòng for duyệt từ chỉ số 0 tới chỉ số i - 1 và kiểm tra xem có giá trị nào bằng với A[i] hay ko, nếu ko có thì hãy in ra, có thể kiểm tra bằng cách sử dụng 1 biến đánh dấu.

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng;

**Constraints**

1<=N<=1000; 0<=A[i], X<=10^3;

**Output Format**

In ra các giá trị khác nhau theo thứ tự xuất hiện, mỗi giá trị chỉ liệt kê 1 lần.

**Sample Input 0**

9

1 2 3 4 1 2 3 4 5

**Sample Output 0**

1 2 3 4 5

# [Mảng 1 chiều cơ bản]. Bài 4. Tần suất

Cho mảng số nguyên A[] gồm N phần tử, hãy liệt kê các giá trị xuất hiện trong mảng kèm theo tần suất tương ứng, mỗi giá trị chỉ liệt kê một lần theo thứ tự xuất hiện.

Gợi ý : Xét từng chỉ số i trong mảng, đối với mỗi chỉ số i sẽ duyệt các phần tử đứng trước nó để xem nó đã xuất hiện trước đó hay chưa, nếu chưa xuất hiện thì tiến hành duyệt các phần tử đứng sau chỉ số i và đếm xem có bao nhiêu phần tử bằng với a[i]

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng;

**Constraints**

1<=N<=1000; 0<=A[i]<=10^3;

**Output Format**

In ra nhiều dòng, mỗi dòng gồm giá trị kèm theo tần suất tương ứng

**Sample Input 0**

7

4 2 6 3 0 7 7

**Sample Output 0**

4 1

2 1

6 1

3 1

0 1

7 2

**Sample Input 1**

5

1 1 1 1 1

**Sample Output 1**

1 5

**Sample Input 2**

6

1 2 3 3 3 3

**Sample Output 2**

1 1

2 1

3 4

# [Mảng 1 chiều cơ bản]. Bài 5. Cân bằng nguyên tố

Cho mảng số nguyên A[] gồm N phần tử, hãy liệt kê các chỉ số i trong mảng thỏa mãn : Tổng các phần tử bên trái i và tổng các phần tử bên phải i là các số nguyên tố

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng;

**Constraints**

1<=N<=1000; 0<=A[i], X<=10^3;

**Output Format**

In ra các chỉ số thỏa mãn trên một dòng, các số cách nhau một khoảng trắng

**Sample Input 0**

5

53 5 69 47 19

**Sample Output 0**

3

# [Mảng 1 chiều cơ bản]. Bài 6. Vị trí số lớn nhất, nhỏ nhất

Cho mảng số nguyên A[] gồm N phần tử, hãy tìm vị trí(bắt đầu từ 0) cuối cùng của giá trị nhỏ nhất trong mảng và vị trí đầu tiên của giá trị lớn nhất trong mảng. Tức là nếu có nhiều số có cùng giá trị nhỏ nhất bạn phải in ra ví trí cuối cùng, và có nhiều số có cùng giá trị lớn nhất trong mảng bạn phải in ra vị trí đầu tiên lớn nhất đó.

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử viết cách nhau một khoảng trống.

**Constraints**

1<=N<=10^6; 1<=A[i]<=10^6

**Output Format**

In trên 1 dòng 2 chỉ số mà bạn tìm được.

**Sample Input 0**

9

936 234 471 168 834 82 280 674 881

**Sample Output 0**

5 0

# [Mảng 1 chiều cơ bản]. Bài 7. Tính tổng và tích các phần tử

Cho mảng số nguyên A[] gồm N phần tử, hãy tính tổng, tích của các phần tử trong mảng và lấy dư với 10^9+7.

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử viết cách nhau một khoảng trống.

**Constraints**

1<=N<=10^6; 0<=A[i]<=10^6

**Output Format**

Dòng đầu tiên in ra tổng các phần tử trong mảng chia dư với 10^9 + 7; Dòng thứ hai in ra tích các phần tử trong mảng chia dư với 10^9 +7;

**Sample Input 0**

6

997893 995053 997553 996212 998316 992144

**Sample Output 0**

5977171

436766709

# [Mảng 1 chiều cơ bản]. Bài 8. Tìm gcd của mọi phần tử trong mảng

Cho mảng số nguyên A[] gồm N phần tử, tìm số lớn nhất mà mọi số trong mảng đều chia hết cho số đó.

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử viết cách nhau một khoảng trống.

**Constraints**

1<=N<=10^6; 0<=A[i]<=10^6

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

12

994626 994448 996097 995790 999692 993222 991401 999038 997163 990016 993004 990549

**Sample Output 0**

1

# [Mảng 1 chiều cơ bản]. Bài 9. Số lớn thứ nhất và lớn thứ 2

Cho mảng số nguyên A[] gồm N phần tử, tìm số lớn nhất và lớn thứ 2 trong mảng. Chú ý 2 giá trị này có thể giống nhau

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử viết cách nhau một khoảng trống.

**Constraints**

2<=N<=10^6; 0<=A[i]<=10^6

**Output Format**

In ra số lớn thứ nhất và lớn thứ 2 trong mảng

**Sample Input 0**

4

996817 993012 990914 998837

**Sample Output 0**

998837 996817

# [Mảng 1 chiều cơ bản]. Bài 10. Mảng đối xứng

Cho mảng số nguyên A[] gồm N phần tử, kiểm tra xem mảng có đối xứng hay không?

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử viết cách nhau một khoảng trống.

**Constraints**

1<=N<=10^6; 0<=A[i]<=10^6

**Output Format**

In ra YES hoặc NO tương ứng với mảng đối xứng hoặc không.

**Sample Input 0**

20

870 1069 2363 375 2188 6059 6775 495 643 832 832 643 495 6775 6059 2188 375 2363 1069 870

**Sample Output 0**

YES

# [Mảng 1 chiều cơ bản]. Bài 18. Liền kề trái dấu

Cho mảng số nguyên A[] gồm N phần tử, hãy liệt kê theo thứ tự xuất hiện các số thỏa mãn có ít nhất 1 số trái dấu với nó đứng cạnh nó.

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử viết cách nhau một khoảng trống.

**Constraints**

2<=N<=10^6; -10^6<=A[i]<=10^6

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

9

-992414 993205 998602 990785 -993397 990527 995429 991288 -992221

**Sample Output 0**

-992414 993205 990785 -993397 990527 991288 -992221

# [Mảng 1 chiều cơ bản]. Bài 19. Lật ngược mảng

Cho mảng số nguyên A[] gồm N phần tử, hãy lật ngược mảng A[] và in ra kết quả

**Input Format**

Dòng đầu tiên là N : số lượng phần tử trong mảng; Dòng thứ 2 gồm N phần tử viết cách nhau một khoảng trống.

**Constraints**

1<=N<=1000; 0<=A[i]<=10^3;

**Output Format**

In ra mảng sau khi đã đảo ngược các phần tử

**Sample Input 0**

5

69 94 77 28 81

**Sample Output 0**

81 28 77 94 69

# [Mảng 1 chiều cơ bản]. Bài 20. Lớn hơn liền kề

Cho mảng số nguyên A[] gồm N phần tử, nhiệm vụ của bạn là liệt kê các phần tử trong mảng thỏa mãn nó lớn hơn cả phần tử đứng trước và phần tử đứng sau nó. 2 phần tử đầu tiên và cuối cùng của mảng không được coi là thỏa mãn.

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; 0<=A[i]<=10^3;

**Output Format**

In ra các số thỏa mãn trên 1 dòng

**Sample Input 0**

7

341 607 986 616 131 857 234

**Sample Output 0**

986 857

# [Mảng 1 chiều cơ bản]. Bài 21. Chèn 1.

Cho mảng số nguyên A[] gồm N phần tử, nhiệm vụ của bạn là chèn phần tử X vào vị trí K trong mảng sau đó in ra mảng.

**Input Format**

Dòng đầu tiên là 3 số nguyên dương N, X, K; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=K<=N<=1000; -10^3<=A[i], X<=10^3;

**Output Format**

In ra mảng sau khi chèn

**Sample Input 0**

8 292 5

343 473 572 525 210 2 874 318

**Sample Output 0**

343 473 572 525 292 210 2 874 318

# [Mảng 1 chiều cơ bản]. Bài 22. Xóa

Cho mảng số nguyên A[] gồm N phần tử và số nguyên X, nhiệm vụ của bạn là tìm kiếm xem X có xuất hiện trong mảng hay không, nếu X xuất hiện trong mảng thì thực hiện xóa vị trí xuất hiện đầu tiên của X trong mảng, ngược lại in ra "NOT FOUND" nếu X không xuất hiện.

**Input Format**

Dòng đầu tiên là 2 số nguyên dương N, X; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; -10^3<=A[i], X<=10^3;

**Output Format**

In ra mảng sau khi xóa hoặc in ra NOT FOUND nếu X không xuất hiện trong mảng

**Sample Input 0**

9 212

359 523 428 952 752 195 613 6 122

**Sample Output 0**

NOT FOUND

**Sample Input 1**

5 3

1 3 3 2 5

**Sample Output 1**

1 3 2 5

# [Mảng 1 chiều cơ bản]. Bài 23. Mảng cộng dồn

Cho mảng số nguyên A[] gồm N phần tử, nhiệm vụ của bạn là xây dựng mảng F cũng có N phần tử trong đó F[0] = A[0] và F[i] = F[i -1] + A[i] với mọi i >= 1. Như vậy bạn thử nghĩ xem F[i] lưu giá trị gì?

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; 0<=A[i]<=10^3;

**Output Format**

In ra mảng cộng dồn F

**Sample Input 0**

5

124 577 658 919 87

**Sample Output 0**

124 701 1359 2278 2365

# [Mảng 1 chiều cơ bản]. Bài 24. Cửa sổ cỡ K

Cho mảng số nguyên A[] gồm N phần tử và số nguyên K, nhiệm vụ của bạn là tìm tổng của mọi dãy con liên tiếp cỡ K của mảng A[]

Gợi ý : Duyệt các chỉ số bắt đầu của các dãy con cỡ K của mảng, chỉ số này chạy từ i = 0, tới i = n - k.

**Input Format**

Dòng đầu tiên là 2 số nguyên dương N, K; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=K<=N<=1000; 0<=A[i]<=10^3;

**Output Format**

In ra tổng của các dãy con K phần tử liên tiếp trong mảng

**Sample Input 0**

7 5

5 9 4 5 5 8 7

**Sample Output 0**

28 31 29

**Explanation 0**

5 + 9 + 4 + 5 + 5 = 28 9 + 4 + 5 + 5 + 8 = 31 4 + 4 + 4 + 8 + 7 = 29

**Sample Input 1**

6 3

3 8 6 8 0 0

**Sample Output 1**

17 22 14 8

# [Mảng 1 chiều cơ bản]. Bài 25. Mảng chẵn lẻ

Cho mảng số nguyên A[] có không quá 10000 phần tử. Hãy xác định xem mảng là mảng chẵn hay mảng lẻ, mảng chẵn được định nghĩa là mảng mà số lượng số chẵn nhiều hơn số lượng số lẻ, ngược lại. Trong trường hợp số lượng số chẵn bằng số lượng số lẻ thì mảng được gọi là mảng chẵn lẻ.

**Input Format**

Gồm nhiều dòng, mỗi dòng có nhiều số là các số trong mảng A[], không biết trước có bao nhiêu số nhưng không vượt quá 10000 số.

**Constraints**

1<=A[i]<=1000

**Output Format**

In ra CHAN nếu mảng chẵn, LE nếu mảng lẻ, CHANLE nếu mảng chẵn lẻ

**Sample Input 0**

3

9 7 9

7 5 5

3

**Sample Output 0**

LE

# [Mảng 1 chiều cơ bản]. Bài 26. Mảng đánh dấu 1

Cho mảng số nguyên A[] gồm N phần tử, hãy đếm xem có bao nhiêu giá trị khác nhau trong mảng?

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; 0<=A[i]<=10^6;

**Output Format**

In ra số lượng các giá trị khác nhau xuất hiện trong mảng

**Sample Input 0**

5

3 9 4 0 9

**Sample Output 0**

4

**Sample Input 1**

5

1 7 8 1 7

**Sample Output 1**

3

# [Mảng 1 chiều cơ bản]. Bài 27. Mảng đánh dấu 2

Cho mảng số nguyên A[] gồm N phần tử, hãy liệt kê các giá trị xuất hiện trong mảng theo thứ tự từ nhỏ đến lớn kèm theo tần suất của nó

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; 0<=A[i]<=10^6;

**Output Format**

In ra nhiều dòng, mỗi dòng gồm giá trị và tần suất viết cách nhau một khoảng trắng

**Sample Input 0**

6

6 6 4 6 0 8

**Sample Output 0**

0 1

4 1

6 3

8 1

# [Mảng 1 chiều cơ bản]. Bài 28. Mảng đánh dấu 3

Cho mảng số nguyên A[] gồm N phần tử, hãy liệt kê các giá trị xuất hiện trong mảng theo thứ tự xuất hiện trong mảng kèm theo tần suất của nó, mỗi giá trị chỉ liệt kê một lần.

Gợi ý : Dùng map hoặc mảng đánh dấu để đếm tần suất sau đó thì duyệt mảng và gặp a[i] thì in ra a[i] kèm tần suất, sau đó có thể xóa a[i] hoặc reset tần suất a[i] về 0 để tránh in trùng

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; 0<=A[i]<=10^6;

**Output Format**

In ra nhiều dòng, mỗi dòng gồm giá trị và tần suất viết cách nhau một khoảng trắng

**Sample Input 0**

9

1 6 6 9 0 6 4 9 3

**Sample Output 0**

1 1

6 3

9 2

0 1

4 1

3 1

# [Mảng 1 chiều cơ bản]. Bài 29. Mảng đánh dấu 4

Cho mảng số nguyên A[] gồm N phần tử, hãy tìm giá trị có số lần xuất hiện nhiều nhất trong mảng, nếu có nhiều giá trị có cùng số lần xuất hiện thì lấy số có giá trị nhỏ nhất

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; 0<=A[i]<=10^6;

**Output Format**

In ra giá trị có số lần xuất hiện nhiều nhất kèm theo tần suất của nó

**Sample Input 0**

5

9 4 0 4 5

**Sample Output 0**

4 2

**Sample Input 1**

5

1 3 7 7 5

**Sample Output 1**

7 2

**Sample Input 2**

5

1 2 3 4 5

**Sample Output 2**

1 1

# [Mảng 1 chiều cơ bản]. Bài 30. Mảng đánh dấu 5

Cho mảng số nguyên A[] gồm N phần tử, hãy tìm giá trị có số lần xuất hiện nhiều nhất trong mảng, nếu có nhiều giá trị có cùng số lần xuất hiện thì lấy giá trị xuất hiện trước theo thứ tự trong mảng

**Input Format**

Dòng đầu tiên là số nguyên dương N; Dòng thứ 2 gồm N số nguyên viết cách nhau một vài khoảng trắng

**Constraints**

1<=N<=1000; 0<=A[i]<=10^6;

**Output Format**

In ra giá trị có số lần xuất hiện nhiều nhất kèm theo tần suất của nó

**Sample Input 0**

6

7 2 0 3 9 5

**Sample Output 0**

7 1

**Sample Input 1**

4

1 2 2 1

**Sample Output 1**

1 2

# [Mảng 1 chiều cơ bản]. Bài 31. Chèn mảng

Cho mảng A[], B[] gồm N và M phần tử, hãy chèn mảng B vào chỉ số P của mảng A và in ra mảng A[] sau khi chèn.

**Input Format**

* Dòng 1 gồm N, M và P
* Dòng 2 gồm các phần trong mảng A[]
* Dòng 3 gồm các phần tử trong mảng B[]

**Constraints**

* 1<=N<=M<=1000
* 0<=P<=N - 1
* 0<=A[i], B[i]<=1000

**Output Format**

* In ra mảng A[] sau khi chèn

**Sample Input 0**

10 7 5

64 94 78 99 50 21 57 41 68 16

73 91 15 89 2 85 61

**Sample Output 0**

64 94 78 99 50 73 91 15 89 2 85 61 21 57 41 68 16

# [Mảng 1 chiều cơ bản]. Bài 32. Xoay trái

Cho mảng A[] gồm N phần tử, hãy dịch các phần tử trong mảng sang trái K vị trí, khi dịch trái thì phần tử đầu tiên của dãy sẽ quay vòng lại trở thành phần tử cuối cùng

**Input Format**

* Dòng 1 gồm N và K
* Dòng 2 gồm N số của mảng A[]

**Constraints**

* 1<=N<=1000
* 1<=K<=1000
* 0<=A[i]<=100

**Output Format**

In ra mảng sau khi xoay trái

**Sample Input 0**

8 6

16 66 96 73 11 13 99 82

**Sample Output 0**

99 82 16 66 96 73 11 13

# [Mảng 1 chiều cơ bản]. Bài 33. Xoay phải

Cho mảng A[] gồm N phần tử, hãy dịch các phần tử trong mảng sang phải K vị trí, khi dịch phải thì phần tử cuối cùng của dãy sẽ quay vòng lại trở thành phần tử đầu tiên

**Input Format**

* Dòng 1 gồm N và K
* Dòng 2 gồm N số của mảng A[]

**Constraints**

* 1<=N<=1000
* 1<=K<=1000
* 0<=A[i]<=100

**Output Format**

In ra mảng sau khi xoay trái

**Sample Input 0**

9 3

69 3 92 57 52 67 30 42 90

**Sample Output 0**

30 42 90 69 3 92 57 52 67

# [Mảng 1 chiều cơ bản]. Bài 34. Unique

Cho mảng A[] gồm N phần tử, nhiệm vụ của bạn là xóa các phần tử trong mảng sao cho không có 2 phần tử liền kề có giá trị giống nhau. Ví dụ mảng A[] = {1, 1, 2, 2, 2, 3, 3, 2, 1, 4} sau khi xóa sẽ được A[] = {1, 2, 3, 2, 1, 4}

**Input Format**

* Dòng 1 gồm N
* Dòng 2 gồm N số của mảng A[]

**Constraints**

* 1<=N<=1000
* 0<=A[i]<=100

**Output Format**

In ra mảng sau khi xóa

**Sample Input 0**

11

2 3 3 0 0 0 3 3 3 3 4

**Sample Output 0**

2 3 0 3 4

# [Mảng 1 chiều cơ bản]. Bài 35. Cộng trừ

Cho mảng A[] gồm N phần tử là các số nguyên, bạn hãy thực hiện phép tính cộng hoặc trừ N số nguyên này theo hướng dẫn. Bạn được cấp 1 mảng B[] gồm N - 1 phần tử đại diện cho N - 1 phép toán giữa N phần tử ban đầu trong mảng, trong đó 1 tương ứng với phép cộng và 2 tương ứng với phép trừ. Ví dụ A[] = {1, 2, 3, 4, 5} và B[] = {1, 1, 2, 1} ta sẽ tính giá trị của mảng A[] = 1 + 2 + 3 - 4 + 5

**Input Format**

* Dòng 1 gồm N
* Dòng 2 gồm N số của mảng A[]
* Dòng 3 gồm N - 1 phần tử của mảng B[]

**Constraints**

* 1<=N<=1000
* 0<=A[i]<=100

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

7

2 1 4 4 1 2 1

2 2 2 1 2 1

**Sample Output 0**

-7

# [Mảng 1 Chiều]. Bài 36. Các bài toán cơ bản trên mảng.

Cho mảng số nguyên gồm N phần tử. Thực hiện in ra các yêu cầu sau đây. (Bài này đúng là được nhé mọi người, TLE không sao cả vì test to quá)

**Input Format**

Dòng đầu tiên là số lượng phần tử trong mảng. Dòng thứ 2 là N phần tử trong mảng.

**Constraints**

2≤n≤10^6; -10^9≤ai≤10^9

**Output Format**

Dòng đầu tiên ghi số lớn nhất trong mảng kèm theo vị trí của nó (bắt đầu từ 0), nếu có nhiều số cùng có giá trị lớn nhất thì lấy vị trí xuất hiện đầu tiên. Dòng thứ 2 ghi số nhỏ nhất trong mảng kèm theo vị trí của nó (bắt đầu từ 0), nếu có nhiều số cùng có giá trị nhỏ nhất thì lấy vị trí xuất hiện cuối cùng. Dòng thứ 3 ghi ra số lượng số nguyên tố xuất hiện trong dãy. Dòng thứ 4 ghi ra tích lớn nhất của 2 số trong mảng (2 số khi nhân với nhau tạo ra tích lớn nhất). Dòng thứ 5 in ra YES nếu mảng đối xứng, ngược lại in ra NO. Dòng thứ 6 tính tích các số trong mảng, kết quả lấy dư với (10^9+7)

**Sample Input 0**

6

-10 -10 2 3 4 5

**Sample Output 0**

5 5

-10 1

3

100

NO

12000

# [Mảng 1 Chiều]. Bài 37. Kiểm tra mảng tăng dần.

Kiểm tra xem mảng đã cho có tăng chặt hay không, tức là các phần tử đứng sau luôn lớn hơn phần tử đứng trước nó.

**Input Format**

Dòng đầu tiên là số lượng phần tử trong mảng. Dòng thứ 2 là N phần tử trong mảng.

**Constraints**

1≤n≤10^6; 1≤ai≤10^9

**Output Format**

In YES nếu các phần tử trong mảng tăng dần, ngược lại in NO.

**Sample Input 0**

6

1 2 3 8 9 22

**Sample Output 0**

YES

**Sample Input 1**

5

1 2 8 8 10

**Sample Output 1**

NO

# [Mảng 1 Chiều].Bài 38. Số lớn hơn các số đứng trước

Cho một dãy số nguyên dương có n phần tử. Hãy liệt kê số các phần tử trong dãy lớn hơn tất cả các số đứng trước nó (Phần tử đầu tiên được coi là một phần tử thỏa mãn).

**Input Format**

Dòng đầu tiên là số lượng phần tử trong mảng. Dòng thứ 2 là N phần tử trong mảng.

**Constraints**

2≤n≤10^6; 1≤ai≤10^9

**Output Format**

Liệt kê các số thỏa mãn

**Sample Input 0**

6

2 6 1 3 9 9

**Sample Output 0**

2 6 9

# [Mảng 1 Chiều].Bài 39. Die hard

Bộ phim "Die Hard" mới vừa được phát hành! Có n người tại phòng vé rạp chiếu phim đứng thành một hàng lớn. Mỗi người trong số họ có một tờ tiền mệnh giá 100, 50 hoặc 25 rúp. Một vé "Die Hard" có giá 25 rúp. Nhân viên đặt phòng có thể bán vé cho mỗi người và trả tiền thừa nếu ban đầu anh ta không có tiền và bán vé theo đúng thứ tự mọi người trong hàng không?

**Input Format**

Dòng đầu tiên chứa số nguyên n - số người trong hàng. Dòng tiếp theo chứa n số nguyên, mỗi số bằng 25, 50 hoặc 100 - giá trị của các tờ tiền mà mọi người có

**Constraints**

1≤n≤10^6

**Output Format**

In YES nếu người bán hàng có thể bán và trả tiền thừa cho mọi người trong hàng, ngược lại in NO

**Sample Input 0**

5

25 25 25 50 50

**Sample Output 0**

YES

**Sample Input 1**

2

50 25

**Sample Output 1**

NO

# [Mảng 1 Chiều]. Bài 40. Gửi thư

Tất cả các thành phố của Lineland đều nằm trên trục tọa độ Ox. Do đó, mỗi thành phố được liên kết với vị trí xi - tọa độ trên trục Ox. Không có hai thành phố được đặt tại một điểm. Cư dân Lineland thích gửi thư cho nhau. Một người chỉ có thể gửi thư nếu người nhận sống ở một thành phố khác. Chi phí gửi thư chính xác bằng khoảng cách giữa thành phố của người gửi và thành phố của người nhận. Đối với mỗi thành phố, hãy tính hai giá trị mini và maxi, trong đó mini là chi phí tối thiểu để gửi thư từ thành phố thứ i đến một thành phố khác và maxi là chi phí tối đa để gửi thư từ thành phố thứ i đến một số thành phố khác

**Input Format**

Dòng đầu tiên là số nguyên dương n Dòng thứ hai chứa chuỗi n số nguyên khác nhau x1, x2, ..., xn (-10^9<= xi <=10^9), trong đó xi là tọa độ x của thành phố thứ i. Tất cả các xi là khác biệt và theo thứ tự tăng dần.

**Constraints**

2 ≤ n ≤ 10^6; -10^9 ≤ xi ≤ 10^9

**Output Format**

Đối với mỗi thành phố in ra 2 giá trị mini và maxi trên 1 dòng.

**Sample Input 0**

4

-5 -2 2 7

**Sample Output 0**

3 12

3 9

4 7

5 12

# [Mảng 1 Chiều]. Bài 41. Đếm cặp số nguyên tố cùng nhau

Cho một dãy số nguyên dương có n phần tử. Hãy đếm các cặp số nguyên tố cùng nhau trong mảng.

**Input Format**

Dòng đầu tiên là số lượng phần tử trong mảng n. Dòng thứ 2 là các phần tử ai trong mảng

**Constraints**

1≤n≤1000; 1≤ai≤10^9

**Output Format**

In ra số lượng cặp số nguyên tố cùng nhau trong mảng.

**Sample Input 0**

5

2 4 8 3 6

**Sample Output 0**

3

**Explanation 0**

Các cặp số nguyên tố cùng nhau là : (2, 3), (3, 4), (3, 8)

# [Mảng 1 Chiều]. Bài 42. Sắp xếp chẵn lẻ.

Cho một mảng các số nguyên, sắp xếp các phần tử trong mảng sao cho, các phần tử lẻ đứng trước và giảm dần, các phần tử chẵn đứng sau và tăng dần. Xem thêm ví dụ để hiểu rõ hơn yêu cầu.

**Input Format**

Dòng đầu tiên là số lượng phần tử trong mảng n Dòng thứ 2 là các phần tử ai trong mảng

**Constraints**

1≤n≤10^6; 1≤ai≤10^9

**Output Format**

In ra dãy đã được sắp xếp

**Sample Input 0**

10

1 2 3 9 7 4 8 6 10 5

**Sample Output 0**

9 7 5 3 1 2 4 6 8 10

# [Mảng 1 Chiều]. Bài 43. Trộn 2 dãy đã sắp xếp

Cho 2 mảng đã được sắp xếp tăng dần, thực hiện trộn 2 dãy trên thành một dãy được sắp xếp.

**Input Format**

Dòng đầu tiên là số lượng phần tử của 2 dãy n và m Dòng thứ 2 là n phần tử trong dãy số 1. Dòng thứ 3 là m phần tử trong dãy thứ 2.

**Constraints**

1≤n,m≤10^7; 1≤ai≤10^9

**Output Format**

In ra mảng sau khi trộn

**Sample Input 0**

42 469

35 145 153 169 281 292 299 322 333 334 358 382 391 421 436 447 464 478 491 500 538 604 667 703 705 716 718 724 726 771 811 827 827 869 894 895 902 912 942 961 962 995

2 3 3 6 7 8 9 10 11 18 18 19 20 21 21 21 22 23 24 30 30 31 31 35 37 38 40 41 43 44 52 53 53 55 60 60 61 64 70 72 75 82 84 86 87 88 93 97 99 101 102 105 106 107 109 114 115 116 118 119 123 125 127 129 139 140 141 142 142 144 145 150 152 153 154 154 156 157 159 161 161 164 168 168 169 170 170 173 177 179 181 182 185 186 187 189 190 190 191 191 192 193 195 195 195 196 199 200 202 202 202 209 213 221 222 222 224 227 235 249 253 253 255 260 261 263 264 264 270 270 270 272 279 281 281 282 285 285 286 286 287 288 290 291 292 292 297 297 302 303 306 308 309 310 313 313 313 314 314 315 316 316 321 322 323 326 328 329 334 337 343 348 350 350 350 355 355 359 360 362 368 370 371 372 374 376 383 386 389 391 393 401 410 411 413 413 414 416 416 418 421 422 423 423 423 425 426 430 432 433 434 437 439 439 441 446 448 450 451 455 457 458 464 466 466 467 472 474 474 476 477 480 481 483 483 484 484 485 487 487 488 492 503 505 506 510 510 511 512 512 514 518 519 520 520 523 525 527 529 529 535 537 537 538 538 541 543 547 547 548 548 549 549 550 555 556 557 558 565 573 574 576 576 577 580 585 587 588 589 591 593 593 595 596 596 600 600 602 609 616 617 617 618 622 623 624 624 625 625 625 626 627 627 629 629 634 636 639 644 646 646 648 648 649 651 655 655 657 658 658 659 662 664 667 668 668 671 673 673 678 687 688 692 694 694 695 699 701 702 704 711 711 712 718 721 723 724 725 728 729 734 734 737 741 745 745 753 753 756 756 757 757 757 758 758 760 762 763 767 771 773 777 778 786 787 788 789 790 796 798 798 798 800 802 805 807 808 813 815 824 829 832 832 833 833 833 836 840 842 844 844 851 859 861 866 868 869 869 874 875 881 882 886 888 890 892 893 896 900 900 900 902 905 909 912 924 924 926 928 929 930 931 932 935 938 941 942 944 945 945 946 949 954 958 958 962 966 966 972 974 976 977 982 985 986 989 996 998 999

**Sample Output 0**

2 3 3 6 7 8 9 10 11 18 18 19 20 21 21 21 22 23 24 30 30 31 31 35 35 37 38 40 41 43 44 52 53 53 55 60 60 61 64 70 72 75 82 84 86 87 88 93 97 99 101 102 105 106 107 109 114 115 116 118 119 123 125 127 129 139 140 141 142 142 144 145 145 150 152 153 153 154 154 156 157 159 161 161 164 168 168 169 169 170 170 173 177 179 181 182 185 186 187 189 190 190 191 191 192 193 195 195 195 196 199 200 202 202 202 209 213 221 222 222 224 227 235 249 253 253 255 260 261 263 264 264 270 270 270 272 279 281 281 281 282 285 285 286 286 287 288 290 291 292 292 292 297 297 299 302 303 306 308 309 310 313 313 313 314 314 315 316 316 321 322 322 323 326 328 329 333 334 334 337 343 348 350 350 350 355 355 358 359 360 362 368 370 371 372 374 376 382 383 386 389 391 391 393 401 410 411 413 413 414 416 416 418 421 421 422 423 423 423 425 426 430 432 433 434 436 437 439 439 441 446 447 448 450 451 455 457 458 464 464 466 466 467 472 474 474 476 477 478 480 481 483 483 484 484 485 487 487 488 491 492 500 503 505 506 510 510 511 512 512 514 518 519 520 520 523 525 527 529 529 535 537 537 538 538 538 541 543 547 547 548 548 549 549 550 555 556 557 558 565 573 574 576 576 577 580 585 587 588 589 591 593 593 595 596 596 600 600 602 604 609 616 617 617 618 622 623 624 624 625 625 625 626 627 627 629 629 634 636 639 644 646 646 648 648 649 651 655 655 657 658 658 659 662 664 667 667 668 668 671 673 673 678 687 688 692 694 694 695 699 701 702 703 704 705 711 711 712 716 718 718 721 723 724 724 725 726 728 729 734 734 737 741 745 745 753 753 756 756 757 757 757 758 758 760 762 763 767 771 771 773 777 778 786 787 788 789 790 796 798 798 798 800 802 805 807 808 811 813 815 824 827 827 829 832 832 833 833 833 836 840 842 844 844 851 859 861 866 868 869 869 869 874 875 881 882 886 888 890 892 893 894 895 896 900 900 900 902 902 905 909 912 912 924 924 926 928 929 930 931 932 935 938 941 942 942 944 945 945 946 949 954 958 958 961 962 962 966 966 972 974 976 977 982 985 986 989 995 996 998 999

# [Mảng 1 Chiều]. Bài 44. Đếm tần suất

Cho mảng các số nguyên không âm gồm n phần tử, thực hiện đếm tần suất xuất hiện của các phần tử và in theo mẫu.

**Input Format**

Dòng đầu tiên là số lượng phần tử trong mảng. Dòng thứ 2 là N phần tử trong mảng.

**Constraints**

2≤n≤10^6; 0≤ai≤10^7

**Output Format**

In ra tần suất xuất hiện của các phần tử theo thứ tự từ nhỏ tới lớn sau đó bỏ trống 1 dòng và in ra tần suất xuất hiện của các phần tử theo thứ tự xuất hiện trong mảng(mỗi giá trị chỉ liệt kê 1 lần).

**Sample Input 0**

8

2 1 2 3 4 8 2 3

**Sample Output 0**

1 1

2 3

3 2

4 1

8 1

2 3

1 1

3 2

4 1

8 1

# [Mảng 1 Chiều]. Bài 45. Tìm hợp và giao của 2 mảng 1

Cho 2 mảng số nguyên a và b gồm n và m phần tử. Gọi mảng c và d lần lượt là mảng chỉ bao gồm các phần tử khác nhau của a và b. Hãy tìm mảng giao và hợp của mảng c và d và liệt kê theo thứ tự tăng dần.

Hướng dẫn tìm giao : Dùng 1 map để đánh dấu những giá trị xuất hiện trong mảng a, Dùng 1 map để đánh dấu những giá trị xuất hiện trong mảng b. Duyệt 1 trong 2 map và kiểm tra xem phần tử của map hiện tại có nằm trong map thứ 2 hay không => giao. Tìm hợp : Đưa các phần tử trong 2 mảng vào 1 cái set

**Input Format**

Dòng đầu tiên là số lượng phần tử của 2 dãy n và m. Dòng thứ 2 là n phần tử trong dãy số 1. Dòng thứ 3 là m phần tử trong dãy thứ 2.

**Constraints**

1≤n,m≤10^6; 0≤ai≤10^7

**Output Format**

Dòng đầu tiên in ra giao của 2 mảng c và d. Dòng thứ 2 in ra hợp của 2 mảng c và d.

**Sample Input 0**

5 6

1 2 1 2 7

1 2 3 4 5 6

**Sample Output 0**

1 2

1 2 3 4 5 6 7

# [Mảng 1 Chiều]. Bài 46. Tìm hợp và giao của 2 mảng 2

Cho 2 mảng số nguyên a và b gồm n và m phần tử, các phần tử trong mảng là đôi một khác nhau, các phần tử trong 2 mảng đã được sắp xếp theo thứ tự tăng dần. Hãy tìm mảng giao và mảng hợp của 2 mảng.

**Input Format**

Dòng đầu tiên là số lượng phần tử của 2 dãy n và m. Dòng thứ 2 là n phần tử trong dãy số 1. Dòng thứ 3 là m phần tử trong dãy thứ 2.

**Constraints**

1≤n,m≤10^7; -10^7≤ai≤10^7

**Output Format**

Dòng đầu tiên in ra mảng hợp của 2 mảng Dòng thứ 2 in ra mảng giao của 2 mảng

**Sample Input 0**

4 5

1 2 3 4

2 3 5 6 7

**Sample Output 0**

1 2 3 4 5 6 7

2 3

# [Mảng 1 Chiều]. Bài 47. Đếm tần suất số nguyên tố

Cho một dãy số nguyên chưa biết trước số lượng phần tử, hãy đếm tần xuất hiện của các số nguyên tố trong dãy và in ra theo thứ tự xuất hiện trong dãy

**Input Format**

Gồm nhiều dòng chưa các số nguyên trong dãy

**Constraints**

Dãy không quá 10000 số nguyên không âm. Các số trong dãy là số nguyên không âm không quá 10^9

**Output Format**

In ra các số nguyên tố trong dãy kèm theo tần suất của nó.

**Sample Input 0**

2 3 5 3 2 5 1

**Sample Output 0**

2 2

3 2

5 2

# [Mảng 1 Chiều]. Bài 48. Đổi tiền tham lam

Tại ngân hàng có các mệnh giá bằng 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, số lượng tờ tiền mỗi mệnh giá là không hạn chế. Một người cần đổi số tiền có giá trị bằng N. Hãy xác định xem số tờ tiền ít nhất sau khi đổi là bao nhiêu?

**Input Format**

Gồm 1 số nguyên N

**Constraints**

1 ≤ N ≤ 10^14

**Output Format**

Số lượng tờ tiền ít nhất có tổng bằng N.

**Sample Input 0**

121

**Sample Output 0**

3

# [Mảng 1 Chiều]. Bài 49. Liệt kê và đếm

Cho một dãy các số nguyên dương không quá 9 chữ số, mỗi số cách nhau vài khoảng trống, có thể xuống dòng. Hãy tìm các số không giảm (các chữ số theo thứ tự từ trái qua phải tạo thành dãy không giảm) và đếm số lần xuất hiện của các số đó.

**Input Format**

Gồm 1 dãy các số nguyên dương không quá 9 chữ số

**Constraints**

Dãy không quá 100000 số. Các số đều nguyên dương và không quá 9 chữ số.

**Output Format**

Ghi ra các số không giảm kèm theo số lần xuất hiện. Các số được liệt kê theo thứ tự sắp xếp số lần xuất hiện giảm dần. Trong trường hợp có nhiều số có cùng số lần xuất hiện thì thì số nhỏ hơn sẽ xếp trước.

**Sample Input 0**

888 289 123

321 54 888

**Sample Output 0**

888 2

123 1

289 1

# [Mảng 1 Chiều]. Bài 50. BRT

Thành phố X có N thị trấn trên trục đường chính. Tọa độ của các thị trấn lần lượt là a[1],a[2], …, a[N], các tọa độ này là phân biệt, không có 2 tọa độ nào trùng nhau. Chính quyền thành phố muốn xây dựng một tuyến buýt nhanh BRT để kết nối 2 thị trấn gần nhau nhất với nhau. Bạn hãy tính thử xem chiều dài của tuyến buýt này bằng bao nhiêu? Và có bao nhiêu cặp thị trấn có tiềm năng giống nhau để xây dựng tuyến BRT này.

**Input Format**

Dòng đầu tiên là số nguyên N (N ≤ 1000 000). Dòng tiếp theo gồm N số nguyên A[i]

**Constraints**

N ≤ 1000 000;-10^9 ≤ A[i] ≤ 10^9

**Output Format**

In ra 2 số nguyên C và D, lần lượt là khoảng cách ngắn nhất giữa 2 thị trấn, và số lượng cặp thị trấn có cùng khoảng cách ngắn nhất này.

**Sample Input 0**

4

6 -3 0 4

**Sample Output 0**

2 1

**Explanation 0**

Khoảng cách nhỏ nhất giữa 2 trị trấn là giữa thị trấn có tọa độ 4 và tọa độ 6.

# [Mảng 1 Chiều]. Bài 51. Định lý Pytago

Theo định lý Pytago, ta đã biết một bộ 3 số (a, b, c) thỏa mãn a^2 + b^2 = c^2 thì đó là ba cạnh của một tam giác vuông. Cho dãy số A[] gồm có N phần tử. Nhiệm vụ của bạn là kiểm tra xem có tồn tại bộ ba số thỏa mãn là ba cạnh của tam giác vuông hay không.

**Input Format**

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

**Constraints**

1≤ N ≤ 5000; 1 ≤ A[i] ≤ 10^9

**Output Format**

In YES nếu trong mảng tồn tại 3 cặp thỏa mãn bộ 3 Pytago, ngược lại in NO.

**Sample Input 0**

3

3 4 5

**Sample Output 0**

YES

# [Mảng 1 Chiều]. Bài 52. Đếm số lượng cặp số bằng nhau trong mảng

Cho dãy số A[] gồm có N phần tử, nhiệm vụ của bạn là đếm số lượng cặp số bằng nhau trong mảng.

**Input Format**

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

**Constraints**

1≤ N ≤ 10^6; 1 ≤ A[i] ≤ 10^6

**Output Format**

In ra số lượng cặp số bằng nhau trong mảng, không xét đến thứ tự.

**Sample Input 0**

6

2 2 1 3 2 3

**Sample Output 0**

4

**Explanation 0**

Các cặp số bằng nhau theo chỉ số : (0, 1), (0, 4), (1, 4), (3, 5)

# [Mảng 1 Chiều]. Bài 53. Dãy con dài nhất các phần tử liền kề khác nhau

Cho dãy số A[] gồm có N phần tử, nhiệm vụ của bạn tìm ra độ dài của dãy con liên tiếp các phần tử sao cho các phần tử liền kề nhau trong dãy con đều khác nhau. Nếu có nhiều dãy con thỏa mãn, hay in ra dãy con có tổng lớn nhất.

**Input Format**

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

**Constraints**

1≤ N ≤ 10^6; 1 ≤ A[i] ≤ 10^6

**Output Format**

In ra đáp án của bài toán trên 2 dòng. Dòng đầu tiên là độ dài của dãy con dài nhất. Dòng thứ 2 là các phần tử trong dãy con đó.

**Sample Input 0**

8

1 2 3 4 4 5 6 7

**Sample Output 0**

4

4 5 6 7

# [Mảng 2 Chiều]. Bài 54. Liệt kê các số nguyên tố trong mảng 2 chiều

Cho ma trận cỡ NxM gồm N hàng, mỗi hàng M cột. Hãy liệt kê các số nguyên tố theo từng hàng trong ma trận.

**Input Format**

Dòng đầu tiên là 2 số N và M. N dòng tiếp theo mỗi dòng có M số.

**Constraints**

1≤n,m≤500; Các phần tử trong ma trận là số dương không quá 10^9.

**Output Format**

In ra các số nguyên tố trong ma trận theo từng dòng

**Sample Input 0**

3 3

1 2 3

4 5 6

7 8 9

**Sample Output 0**

2 3

5

7

# [Mảng 2 Chiều]. Bài 55. Tổng hàng, tổng cột trên ma trận

Cho ma trận cỡ NxM gồm N hàng, mỗi hàng M cột. Hãy tính tổng các phần tử trên từng hàng và tổng các phần tử trên từng cột của ma trận.

**Input Format**

Dòng đầu tiên là 2 số N và M. N dòng tiếp theo mỗi dòng có M số.

**Constraints**

1≤n,m≤200; Các phần tử trong ma trận là số dương không quá 10^9.

**Output Format**

Dòng đầu tiên, in ra N số là tổng của N hàng. Dòng thứ 2 in ra M số là tổng của M cột.

**Sample Input 0**

3 3

1 2 3

4 5 6

7 8 9

**Sample Output 0**

6 15 24

12 15 18

# [Mảng 2 Chiều]. Bài 56. Số lớn nhất và nhỏ nhất trong ma trận

Cho ma trận cỡ NxM gồm N hàng, mỗi hàng M cột. Hãy tìm số nhỏ nhất trong ma trận và số lớn nhất trong ma trận, sau đó liệt kê các vị trí xuất hiện của các phần tử này trong ma trận.

**Input Format**

Dòng đầu tiên là 2 số N và M. N dòng tiếp theo mỗi dòng có M số.

**Constraints**

1≤n,m≤205; Các phần tử trong ma trận là số dương không quá 10^9.

**Output Format**

Dòng đầu tiên in ra số nhỏ nhất trong ma trận, các dòng tiếp theo liệt kê các vị trí của số này trong ma trận. Dòng tiếp theo in ra số lớn nhất trong ma trận, các dòng tiếp theo liệt kê các vị trí của số này trong ma trận. Xem test case mẫu để rõ hơn.

**Sample Input 0**

3 3

1 2 1

4 5 6

7 8 9

**Sample Output 0**

1

1 1

1 3

9

3 3

# [Mảng 2 Chiều]. Bài 57. Số thuận nghịch trong tam giác dưới.

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy đếm các số thuận nghịch thuộc tam giác dưới của ma trận trên.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9.

**Output Format**

In ra số lượng số thuận nghịch trong ma trận.

**Sample Input 0**

3

1 2 3

22 14 56

76 5 4

**Sample Output 0**

4

**Explanation 0**

4 số là số thuận nghịch thuộc tam giác dưới là : 1, 22, 4, 5

# [Mảng 2 Chiều]. Bài 58. In ra ma trận theo mẫu.

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy in ra ma trận theo các mẫu dưới đây.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số.

**Constraints**

1≤n≤500; Các phần tử trong ma trận là số dương không quá 10^9.

**Output Format**

In ra ma trận theo mẫu, xem test case để rõ hơn.

**Sample Input 0**

5

510 745 649 186 313

474 22 168 18 787

905 958 391 202 625

477 414 314 824 334

874 372 159 833 70

**Sample Output 0**

Pattern 1:

510 474 905 477 874

745 22 958 414 372

649 168 391 314 159

186 18 202 824 833

313 787 625 334 70

Pattern 2:

70 833 159 372 874

334 824 314 414 477

625 202 391 958 905

787 18 168 22 474

313 186 649 745 510

Pattern 3:

313 787 625 334 70

186 18 202 824 833

649 168 391 314 159

745 22 958 414 372

510 474 905 477 874

Pattern 4:

313 186 649 745 510

787 18 168 22 474

625 202 391 958 905

334 824 314 414 477

70 833 159 372 874

# [Mảng 2 Chiều]. Bài 59. Hoán vị đường chéo

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy hoán vị các phần tử trên 2 đường chéo chính và phụ của ma trận.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9.

**Output Format**

In ra ma trận sau khi hoán vị 2 đường chéo

**Sample Input 0**

8

519 178 774 271 764 669 193 986

103 481 214 628 803 100 528 626

544 925 24 973 62 182 4 433

506 594 726 32 493 143 223 287

65 901 188 361 414 975 271 171

236 834 712 761 897 668 286 551

141 695 696 625 20 126 577 695

659 303 372 467 679 594 852 485

**Sample Output 0**

986 178 774 271 764 669 193 519

103 528 214 628 803 100 481 626

544 925 182 973 62 24 4 433

506 594 726 493 32 143 223 287

65 901 188 414 361 975 271 171

236 834 668 761 897 712 286 551

141 577 696 625 20 126 695 695

485 303 372 467 679 594 852 659

# [Mảng 2 Chiều]. Bài 60. Hoán vị 2 hàng của ma trận

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy hoán vị các phần tử trên 2 hàng của ma trận.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số. Dòng tiếp theo là 2 hàng u, v cần hoán vị.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9; 1≤u,v≤200

**Output Format**

In ra ma trận sau khi hoán vị 2 hàng.

**Sample Input 0**

3

1 2 3

4 5 6

7 8 9

1 3

**Sample Output 0**

7 8 9

4 5 6

1 2 3

# [Mảng 2 Chiều]. Bài 61. Hoán vị 2 cột của ma trận

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy hoán vị các phần tử trên 2 cột của ma trận.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số. Dòng tiếp theo là 2 cột u, v cần hoán vị.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9; 1≤u,v≤200

**Output Format**

In ra ma trận sau khi hoán vị 2 cột.

**Sample Input 0**

3

1 2 3

4 5 6

7 8 9

1 3

**Sample Output 0**

3 2 1

6 5 4

9 8 7

# [Mảng 2 Chiều]. Bài 62. Đếm các phần tử là số nguyên tố trên đường chéo chính và phụ

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy đếm các số nguyên tố trên đường chéo chính và đường chéo phụ, mỗi phần tử là số nguyên tố thỏa mãn chỉ đếm 1 lần.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9;

**Output Format**

In ra số lượng số nguyên tố trên đường chéo chính và phụ.

**Sample Input 0**

4

1 2 3 4

5 6 7 8

9 10 1 2

2 3 5 7

**Sample Output 0**

3

**Explanation 0**

Các phần tử là số nguyên tố trên đường chéo chính là : 7. Các phần tử là số nguyên tố trên đường chéo phụ là : 2, 7

# [Mảng 2 Chiều]. Bài 63. Đếm các phần tử là số nguyên tố trên đường chéo chính và phụ 2

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy đếm các số nguyên tố trên đường chéo chính và đường chéo phụ, mỗi giá trị là số nguyên tố thỏa mãn chỉ đếm 1 lần.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9;

**Output Format**

In ra số lượng số nguyên tố trên đường chéo chính và phụ.

**Sample Input 0**

3

1 2 3

4 5 6

7 8 9

**Sample Output 0**

3

# [Mảng 2 Chiều]. Bài 64. Sắp xếp các hàng của ma trận

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy sắp xếp các phần tử của từng hàng của ma trận theo thứ tự tăng dần.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9;

**Output Format**

In ra ma trận sau khi sắp xếp

**Sample Input 0**

3

2 7 6

1 2 3

9 8 1

**Sample Output 0**

2 6 7

1 2 3

1 8 9

# [Mảng 2 Chiều]. Bài 65. Sắp xếp các phần tử theo cột

Cho ma trận vuông cỡ NxN gồm N hàng, mỗi hàng N cột. Hãy sắp xếp các phần tử trong ma trận theo cột theo thứ tự tăng dần.

**Input Format**

Dòng đầu tiên là số N. N dòng tiếp theo mỗi dòng có N số.

**Constraints**

1≤n≤200; Các phần tử trong ma trận là số dương không quá 10^9;

**Output Format**

In ra ma trận sau khi đã sắp xếp theo cột tăng dần.

**Sample Input 0**

3

1 2 3

5 5 2

1 4 7

**Sample Output 0**

1 2 2

1 4 3

5 5 7

# [Mảng 2 Chiều]. Bài 66. Nhân 2 ma trận.

Cho ma trận A cỡ NxM, ma trận B cỡ MxP. Hãy tính ma trận tích của A và B.

**Input Format**

Dòng đầu tiên là 3 số N, M, P; N dòng tiếp theo, mỗi dòng M của ma trận A; M dòng tiếp theo, mỗi dòng P số của ma trận B

**Constraints**

1≤N,M,P≤50; Các phần tử trong ma trận là số dương không quá 100000;

**Output Format**

In ra ma trận tích của A và B

**Sample Input 0**

3 4 5

1 2 3 4

5 6 7 8

9 10 11 12

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

**Sample Output 0**

110 120 130 140 150

246 272 298 324 350

382 424 466 508 550

# [Mảng 2 Chiều]. Bài 67. Ma trận xoáy ốc

Xây dựng ma trận xoáy ốc cấp N

**Input Format**

Số nguyên dương N là cấp của ma trận xoáy ốc cần xây dựng.

**Constraints**

1≤N≤100

**Output Format**

In ra ma trận xoáy ốc.

**Sample Input 0**

3

**Sample Output 0**

1 2 3

8 9 4

7 6 5

# [Mảng 2 Chiều]. Bài 68. Ma trận xoáy ốc Fibonacci

In ra ma trận xoáy ốc cấp N, với các số trong ma trận đều là các số trong dãy Fibonacci.

**Input Format**

Số nguyên dương N

**Constraints**

1≤N≤9

**Output Format**

Ma trận xoáy ốc cấp N

**Sample Input 0**

3

**Sample Output 0**

0 1 1

13 21 2

8 5 3

# [Mảng 2 Chiều]. Bài 69. Count Island 1

Cho ma trận nhị phân gồm N hàng và M cột chỉ bao gồm các số 0 và 1. Hãy đếm số lượng miền các số 1 trong ma trận, các ô số 1 được coi là cùng miền nếu chúng có chung cạnh.

**Input Format**

Dòng đầu tiên N và M. N dòng tiếp theo mỗi dòng gồm M phần tử.

**Constraints**

1≤N,M≤50.

**Output Format**

In ra số lượng miền số 1 trong ma trận.

**Sample Input 0**

4 7

0 1 1 0 1 1 0

1 1 1 1 1 1 1

1 0 0 0 0 1 1

1 1 0 1 0 0 0

**Sample Output 0**

2

**Sample Input 1**

8 8

1 1 0 1 1 1 1 0

0 1 1 1 0 0 0 1

1 1 0 0 0 0 1 1

0 1 0 1 0 0 1 1

1 0 0 1 0 0 0 1

0 1 0 1 1 1 1 0

0 0 0 0 1 0 1 1

1 0 0 0 0 0 0 0

**Sample Output 1**

6

# [Mảng 2 Chiều]. Bài 70. Count Island 2

Cho ma trận nhị phân gồm N hàng và M cột chỉ bao gồm các số 0 và 1. Hãy đếm số lượng miền các số 1 trong ma trận, các ô số 1 được coi là cùng miền nếu chúng có chung đỉnh.

**Input Format**

Dòng đầu tiên N và M. N dòng tiếp theo mỗi dòng gồm M phần tử.

**Constraints**

1≤N,M≤50.

**Output Format**

In ra số lượng miền số 1 trong ma trận.

**Sample Input 0**

3 3

1 0 1

0 0 1

1 1 0

**Sample Output 0**

2

# [Mảng 2 Chiều]. Bài 71. Số điểm cực đại

Cho ma trận A các số nguyên có N hàng và M cột. Điểm được coi là cực đại trong ma trận nếu nó lớn hơn tất các số ở các số ở ô xung quanh mà chung đỉnh với nó. Hãy đếm xem trong ma trận có bao nhiêu điểm cực đại.

**Input Format**

Dòng đầu tiên N và M. N dòng tiếp theo mỗi dòng gồm M phần tử.

**Constraints**

1≤N,M≤100; 1≤A[i][j]≤10^9

**Output Format**

In ra số lượng các điểm cực đại của ma trận.

**Sample Input 0**

5 3

1 1 2

1 1 1

1 1 2

2 2 1

2 1 2

**Sample Output 0**

1

**Explanation 0**

Có 1 điểm cực đại là điểm (1, 3)

**Sample Input 1**

3 3

1 2 1

1 5 1

1 0 3

**Sample Output 1**

1

**Explanation 1**

Có 1 điểm cực đại là (2, 2)

# [Mảng 2 Chiều]. Bài 72. Maximum path sum

Cho ma trận A các số nguyên có N hàng và M cột. Tìm đường đi từ ở [1, 1] tới ô [N, M] sao cho tổng các số trên đường đi là lớn nhất có thể, biết rằng ở mỗi bước chỉ có thể đi từ ô hiện tại xuống ô phía dưới hoặc đi sang phải.

**Input Format**

Dòng đầu tiên N và M. N dòng tiếp theo mỗi dòng gồm M phần tử.

**Constraints**

1≤N,M≤100; 1≤A[i][j]≤10^9

**Output Format**

In ra đường đi có tổng lớn nhất.

**Sample Input 0**

3 3

1 2 2

3 10 2

5 7 2

**Sample Output 0**

23

**Explanation 0**

Giải thích : Đường đi được chọn (1, 1) -> (2, 1) -> (2, 2) -> (3, 2) -> (3, 3)

# [Mảng 2 Chiều]. Bài 73. Tìm đường đi.

Cho một ma trận nhị phân có N hàng và M cột, một con chuột bắt đầu từ ô có tọa độ [s, t] và tìm đường đi tới ô [u, v], biết rằng ở mỗi bước con chuột có thể di chuyển từ ô hiện tại sang các ô chung cạnh với ô hiện tại và số ở ô chung cạnh là số 1. Bạn chỉ được đi qua 1 ô đúng 1 lần hãy kiểm tra xem con chuột có thể tìm được đường đi tới ô [u, v] hay không ? Dữ liệu đảm bảo 2 ô [s, t] và ô [u, v] đều bằng 1.

**Input Format**

Dòng đầu tiên N và M. Dòng thứ 2 là 4 số s, t, u , v. N dòng tiếp theo mỗi dòng gồm M phần tử.

**Constraints**

1≤N,M≤100; 1≤s,u≤N;1≤t,v≤M;

**Output Format**

In YES nếu con chuột có thể tìm được đường đi, ngược lại in NO.

**Sample Input 0**

3 5

1 1 3 5

1 0 1 0 0

0 1 1 1 1

1 0 0 0 1

**Sample Output 0**

NO

# [Mảng 2 Chiều]. Bài 74. Đường đi của quân Mã

Cho bàn cờ vua cỡ N \* N, các ô trên bàn cờ có giá trị là 0 hoặc 1. Một con mã xuất phát từ ô (s, t) và muốn di chuyển tới ô (u, v), con mã chỉ có thể di chuyển ở các ô mà tại ô đó có giá trị là 1 và nó có thể di chuyển qua lại 1 ô nhiều lần. Hãy xác định xem con mã có thể tìm được đường đi hay không, dữ liệu đảm bảo ô (s, t) và ô (u, v) đều có giá trị là 1.

**Input Format**

Dòng đầu tiên N. Dòng thứ 2 là 4 số s, t, u , v. N dòng tiếp theo mỗi dòng gồm N phần tử.

**Constraints**

1≤N,M≤100; 1≤s,t,u,v≤N; 0≤A[i][j]≤1;

**Output Format**

In YES nếu con mã có thể tìm được đường đi, ngược lại in NO.

**Sample Input 0**

9

7 5 4 3

1 0 1 0 1 0 1 1 1

1 1 1 1 0 0 0 0 1

1 0 1 1 1 0 1 1 1

1 0 1 0 1 0 0 0 0

0 1 1 0 1 1 0 1 1

1 0 0 0 0 1 1 0 1

1 0 1 0 1 0 1 1 0

0 1 1 0 0 0 0 1 1

0 0 1 1 0 0 0 0 1

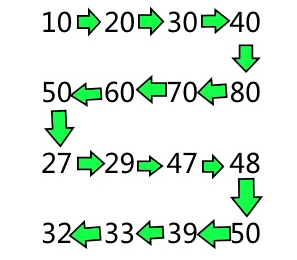
**Sample Output 0**

YES

**Explanation 0**

Con Mã có thể di chuyển từ ô (7, 5) -> (6, 7) -> (5, 5) -> (4, 3)

# [Mảng 2 Chiều]. Bài 75. Snake

Cho ma trận vuông A[N][N]. Hãy in các phần tử thuộc theo hình con rắn.

**Input Format**

Dòng đầu tiên đưa vào N là cấp của ma trận A[N][N]; N dòng tiếp mỗi dòng gồm N số nguyên.

**Constraints**

1≤N≤100; 0≤A[i][j]≤1000000

**Output Format**

In ra ma trận theo yêu cầu.

**Sample Input 0**

15

957 275 516 118 593 6 646 52 910 183 702 335 468 704 506

171 168 428 372 722 683 545 853 409 575 727 930 67 556 753

5 749 378 996 374 171 159 86 100 657 467 609 406 195 570

448 679 401 417 81 954 885 685 316 986 95 297 755 934 107

663 474 203 49 825 440 809 262 463 542 170 929 299 64 473

481 876 994 896 765 632 734 764 691 94 947 925 213 786 735

8 616 394 904 199 775 305 935 797 193 138 598 234 398 480

131 645 527 411 302 627 110 765 880 203 972 445 93 892 746

140 571 401 810 222 515 932 615 252 679 856 58 552 573 595

343 896 35 908 967 862 427 738 199 655 961 678 856 475 435

273 968 573 576 136 274 356 340 263 473 912 819 820 787 576

44 486 724 324 640 326 799 191 378 788 773 31 289 677 71

424 676 915 830 422 934 550 654 274 435 450 165 190 833 601

906 356 74 834 484 892 217 41 625 628 722 801 561 438 816

768 945 804 545 827 732 375 236 57 199 424 647 212 260 110

**Sample Output 0**

957 275 516 118 593 6 646 52 910 183 702 335 468 704 506

753 556 67 930 727 575 409 853 545 683 722 372 428 168 171

5 749 378 996 374 171 159 86 100 657 467 609 406 195 570

107 934 755 297 95 986 316 685 885 954 81 417 401 679 448

663 474 203 49 825 440 809 262 463 542 170 929 299 64 473

735 786 213 925 947 94 691 764 734 632 765 896 994 876 481

8 616 394 904 199 775 305 935 797 193 138 598 234 398 480

746 892 93 445 972 203 880 765 110 627 302 411 527 645 131

140 571 401 810 222 515 932 615 252 679 856 58 552 573 595

435 475 856 678 961 655 199 738 427 862 967 908 35 896 343

273 968 573 576 136 274 356 340 263 473 912 819 820 787 576

71 677 289 31 773 788 378 191 799 326 640 324 724 486 44

424 676 915 830 422 934 550 654 274 435 450 165 190 833 601

816 438 561 801 722 628 625 41 217 892 484 834 74 356 906

768 945 804 545 827 732 375 236 57 199 424 647 212 260 110

**Sample Input 1**

3

1 2 3

4 5 6

7 8 9

**Sample Output 1**

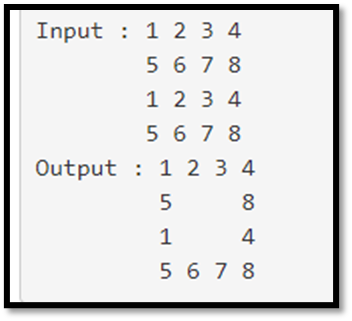
1 2 3

6 5 4

7 8 9

# [Mảng 2 Chiều]. Bài 76. Biên của ma trận

Cho ma trận vuông A[N][N]. Hãy in các phần tử thuộc vùng biên.



**Input Format**

Dòng đầu tiên đưa vào N là cấp của ma trận A[N][N]; N dòng tiếp mỗi dòng gồm N số nguyên.

**Constraints**

1≤N≤100; 0≤A[i][j]≤1000000

**Output Format**

In ra ma trận theo yêu cầu, các số được in trên 1 dòng.

**Sample Input 0**

3

1 2 3

4 5 6

7 8 9

**Sample Output 0**

1 2 3 4 6 7 8 9

# [Mảng 2 Chiều]. Bài 77. Phần tử xuất hiện ở mọi hàng.

Cho ma trận vuông A[N][N]. Hãy in ra các số xuất hiện ở mọi hàng theo thứ tự từ bé đến lớn, mỗi giá trị xuất hiện chỉ liệt kê 1 lần.

**Input Format**

Dòng đầu tiên đưa vào N là cấp của ma trận A[N][N]; N dòng tiếp mỗi dòng gồm N số nguyên.

**Constraints**

1≤N≤100; 0≤A[i][j]≤100

**Output Format**

In ra các giá trị xuất hiện trên mọi dòng của ma trận, các số in cách nhau một dấu cách. Nếu không tồn tại giá trị nào thỏa mãn in ra "NOT FOUND".

**Sample Input 0**

4

1 2 2 3

2 1 2 4

1 4 5 2

2 2 3 1

**Sample Output 0**

1 2

# [Mảng 2 Chiều]. Bài 78. Diện tích

Cho một ma trận nhị phân cỡ NxM chỉ bao gồm các số 0, 1. Hãy xác định vùng nhiều số 1 nhất trên ma trận, biết rằng các ô có chung cạnh được coi là cùng miền.

**Input Format**

Dòng đầu tiên là N và M tương ứng với số hàng, cột của ma trận. N dòng tiếp theo mỗi dòng có M số

**Constraints**

1<=N, M <= 500; Các phần tử trong ma trận có giá trị 0 hoặc 1.

**Output Format**

In ra diện tích miền có nhiều số 1 nhất

**Sample Input 0**

5 6

0 1 0 1 0 1

0 0 0 1 1 0

0 1 1 0 1 1

0 1 1 0 0 0

0 0 0 0 0 0

**Sample Output 0**

5

# [Mảng 2 Chiều]. Bài 79. Maximum path sum 2

Cho mảng 2 chiều A gồm N hàng và N cột, hãy tìm 1 đường đi từ dòng 1 tới dòng N sao cho các phần tử trên đường đi đó là lớn nhất có thể. Biết rằng ở mỗi bước đi từ ô hiện tại chỉ có thể đi xuống ô dưới bên trái, ô dưới bên phải hoặc ô dưới của ô hiện tại. Hãy tìm 1 đường đi có tổng các số trên đường đi là lớn nhất.

**Input Format**

* Dòng 1 là N
* N dòng tiếp theo mỗi dòng gồm N số

**Constraints**

* 1<=N<=100
* 1<=A[i][j]<=100

**Output Format**

In ra kết quả của bài toán

**Sample Input 0**

5

5 1 8 6 7

6 0 0 3 7

3 4 1 5 3

9 8 9 2 2

5 3 4 5 3

**Sample Output 0**

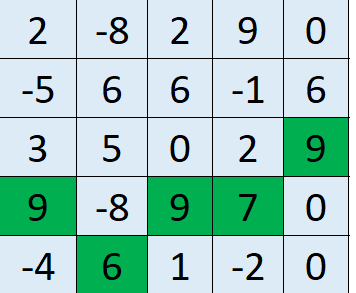
33

**Explanation 0**

Đường đi có tổng lớn nhất : 7 + 7 + 5 + 9 + 5 = 33

# [Mảng 2 Chiều]. Bài 80. Maximum path sum 3

Cho mảng 2 chiều A gồm N hàng và N cột, hàng được đánh số từ 1 đến N từ trên xuống dưới, cột được đánh số từ 1 tới N từ trái sang phải, hãy tìm 1 đường đi từ một ô ở cột 1 tới 1 ổ ở cột N sao cho các phần tử trên đường đi đó là lớn nhất có thể. Biết rằng ở mỗi bước đi từ ô hiện tại (i, j) chỉ có thể đi sang phải (i, j + 1) hoặc đi xuống ô dưới bên phải (i + 1, j + 1), ô bên trên bên phải(i - 1, j + 1). Hãy tìm 1 đường đi có tổng các số trên đường đi là lớn nhất.



**Input Format**

* Dòng 1 là N
* N dòng tiếp theo mỗi dòng gồm N số

**Constraints**

* 1<=N<=100
* -100<=A[i][j]<=100

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

5

2 -8 2 9 0

-5 6 6 -1 6

3 5 0 2 9

9 -8 9 7 0

-4 6 1 -2 0

**Sample Output 0**

40

**Explanation 0**

Đường đi có tổng lớn nhất : 9 + 6 + 9 + 7 + 9 = 40

# [Mảng 2 Chiều]. Bài 81. Hình vuông lớn nhất

Cho mảng 2 chiều A gồm N hàng và N cột, hãy xét tất cả các hình vuông con cỡ 3x3 của mảng A, hãy tìm hình vuông có có tổng lớn nhất đó.

**Input Format**

* Dòng 1 là N
* N dòng tiếp theo mỗi dòng gồm N số

**Constraints**

* 3<=N<=100
* 100<=A[i][j]<=100

**Output Format**

In ra tổng lớn nhất của hình vuông cỡ 3x3

**Sample Input 0**

6

4 6 4 5 8 2

8 0 9 0 1 4

3 5 3 6 3 4

5 7 0 9 5 9

4 5 5 9 6 3

9 2 3 6 1 8

**Sample Output 0**

56

# [Mảng 2 Chiều]. Bài 82. Đường đi của quân Tịnh

Tèo khá thích chơi cờ vua và quân cờ mà Tèo yêu thích chính là quân Tịnh, bây giờ Tèo có một bàn cờ cỡ NxN, trên bàn cờ sẽ có những ô trống và có những ô có vật cản, nhiệm vụ của bạn là hãy xác định xem số lượng ô trên bàn cờ mà quân Tịnh có thể di chuyển tới, biết rằng nó có thể đi qua đi lại 1 ô trống nhiều lần và không thể đi vào ô có vật cản.

Bàn cờ gồm N hàng N cột, mỗi ô là số 1 tương ứng với vật cản và ô số 0 tương ứng với ô trống. Ban đầu quân Tịnh nằm ở vị trí hàng S và cột T và ô (S, T) là ô trống

**Input Format**

* Dòng 1 là N, S, T
* N dòng tiếp theo mỗi dòng gồm N số

**Constraints**

* 5<=N<=20
* 0<=A[i][j]<=1

**Output Format**

In ra số lượng ô trên bàn cờ mà quân Tịnh có thể đến được

**Sample Input 0**

6 1 3

0 0 0 1 1 0

1 1 1 0 1 0

1 0 0 1 0 1

1 1 1 1 1 0

0 0 1 1 1 0

0 1 1 0 1 0

**Sample Output 0**

6

# [Mảng 2 Chiều]. Bài 83. Đường đi của quân Xe

Tèo khá thích chơi cờ vua và quân cờ mà Tèo yêu thích chính là quân Xe, bây giờ Tèo có một bàn cờ cỡ NxN, trên bàn cờ sẽ có những ô trống và có những ô có vật cản, nhiệm vụ của bạn là hãy xác định xem số lượng ô trên bàn cờ mà quân Xe có thể di chuyển tới, biết rằng nó có thể đi qua đi lại 1 ô trống nhiều lần và không thể đi vào ô có vật cản.

Bàn cờ gồm N hàng N cột, mỗi ô là số 1 tương ứng với vật cản và ô số 0 tương ứng với ô trống. Ban đầu quân Xe nằm ở vị trí hàng S và cột T và ô (S, T) là ô trống

**Input Format**

* Dòng 1 là N, S, T
* N dòng tiếp theo mỗi dòng gồm N số

**Constraints**

* 5<=N<=20
* 0<=A[i][j]<=1

**Output Format**

In ra số lượng ô trên bàn cờ mà quân Xe có thể đến được

**Sample Input 0**

7 4 6

1 1 1 0 0 1 0

1 1 0 0 0 1 1

0 1 0 0 0 0 0

1 0 1 0 1 0 1

1 0 1 1 1 1 0

1 1 0 0 1 0 1

0 0 0 0 1 1 0

**Sample Output 0**

12

**Explanation 0**

Những ô quân Xe có thể đi tới : 4 6, 3 6, 3 5, 2 5, 1 5, 1 4, 2 4, 3 4, 4 4, 3 3, 2 3, 3 7

# [Vector]. Bài 84. Vector và pair

Cho N điểm trong hệ tọa độ Oxy, bạn hãy dùng vector

pair<int, int>

để lưu tọa độ các điểm này. Sau đó duyệt vector và tính khoảng cách từ các điểm này về gốc tọa độ và lưu vào 1 vector sau đó in ra các phần tử trong vector khoảng cách này lấy 2 số sau dấu phẩy.

**Input Format**

Dòng 1 là N : số lượng điểm. N dòng tiếp theo mỗi dòng gồm 2 số là tung độ và hoành độ.

**Constraints**

1<=N<=1000; Tọa độ là số nguyên có trị tuyệt đối không quá 100;

**Output Format**

In ra đáp án của bài toán.

**Sample Input 0**

13

5 27

69 84

92 51

12 70

26 70

76 47

36 94

55 2

26 20

85 97

17 31

33 88

10 75

**Sample Output 0**

27.46 108.71 105.19 71.02 74.67 89.36 100.66 55.04 32.80 128.97 35.36 93.98 75.66

# [Vector]. Bài 85. Vector và pair 2

Cho N điểm trong hệ tọa độ Oxyz, bạn hãy dùng vector

pair<pair<int, int>, int>>

để lưu tọa độ các điểm này. Sau đó duyệt vector và in ra tổng của tung độ, hoành độ, cao độ.

**Input Format**

Dòng 1 là N : số lượng điểm. N dòng tiếp theo mỗi dòng gồm 3 số là tung độ, hoành độ, cao độ.

**Constraints**

1<=N<=1000; Tọa độ là số nguyên có trị tuyệt đối không quá 100;

**Output Format**

In ra đáp án của bài toán

**Sample Input 0**

12

65 91 53

64 70 15

50 9 57

69 37 11

31 35 66

73 55 50

63 40 38

33 5 41

67 39 29

85 78 6

67 49 83

41 34 88

**Sample Output 0**

209 149 116 117 132 178 141 79 135 169 199 163