Làm quen với Set, Map Trong C++ Bài

1. Sử dụng cả set và map.

Đếm số lượng phần tử khác nhau trong mảng số nguyên

Input

Dòng đầu tiên là số lượng test case T . ($1 \le T \le 100$).

Mỗi test case bao gồm 2 dòng, dòng đầu tiên là số lượng phần tử trong mảng. ($1 \le n \le 1000$).

Dòng thứ 2 bao gồm n phần tử trong mảng. ($-10^9 \le ai \le 10^9$).

Output

In ra số lượng phần tử khác nhau trong mảng.

Ví dụ

Input	Output
2 5 1 2 2 2 1 4 1 2 3 4	2 4

Bài 2. Sử dụng cả set và map

Cho một mảng số nguyên gồm n phần tử, với mỗi truy vấn hãy kiểm tra xem một số nào đó có nằm trong mảng hay không ?

Input

Dòng đầu tiên là số lượng test case T . ($1 \le T \le 100$).

Mỗi test case bao gồm nhiều dòng, dòng đầu tiên là số lượng phần tử trong mảng. ($1 \le n \le 1000$).

Dòng thứ 2 bao gồm n phần tử trong mảng. ($-10^9 \le ai \le 10^9$).

Dòng thứ 3 là số lượng truy vấn q .

Q dòng tiếp theo mỗi dòng là một số nguyên cần kiểm tra

Output

In ra YES nếu số cần kiểm tra xuất hiện trong mảng, ngược lại in NO Ví dụ

Input	Output
1 6 1 2 3 8 7 0 4 1 2 10 20	YES YES NO NO

Bài 3. Sử dụng cả set và map

Cho 2 mảng số nguyên, hãy đếm những phần tử thuộc mảng thứ nhất mà không thuộc mảng thứ 2.

Input

Dòng đầu tiên là số lượng test case T . ($1 \le T \le 100$).

Mỗi test case gồm 3 dòng, dòng đầu tiên là số lượng phần tử của mảng thứ nhất và mảng thứ 2. ($1 \le n, m \le 1000$).

Dòng thứ 2 là các số thuộc mảng thứ 1. ($-10^9 \le ai \le 10^9$)

Dòng thứ 3 là các số thuộc mảng thứ 2. (($-10^9 \le ai \le 10^9$)

Output

Liệt kê các số thuộc mảng thứ 1 mà không thuộc mảng thứ 2, theo thứ tự xuất hiện, nếu không tồn tại số nào in ra “NOT FOUND”.

Ví dụ

Input	Output
1 4 5 1 2 2 9 2 0 9 8 3	1

Bài 4. Tìm số xuất hiện nhiều lần nhất trong mảng.

Tìm từ xuất hiện nhiều nhất trong mảng, trong trường hợp có nhiều số có cùng số lần xuất hiện thì lấy số nhỏ nhất.

Input

Dòng đầu tiên là số lượng test case T . ($1 \le T \le 100$).

Mỗi test case bao gồm nhiều dòng, dòng đầu tiên là số lượng phần tử trong mảng. ($1 \le n \le 100000$).

Dòng thứ 2 bao gồm n phần tử trong mảng. ($-10^{18} \le ai \le 10^{18}$).

Output

In ra số xuất hiện nhiều nhất cùng số lần xuất hiện của nó.

Ví dụ

Input	Output
1 10 1 1 2 2 2 1 4 7 8 19	 1 3

Bài 5. Tìm kí tự xuất hiện nhiều nhất trong chuỗi.

Sử dụng mảng đếm và sử dụng map

Input

Dòng đầu tiên là số lượng test case T . ($1 \le T \le 100$).

Mỗi test case gồm một dòng là 1 chuỗi có không quá 100000 kí tự, bao gồm cả dấu cách. Output

Tìm kí tự có số lần xuất hiện nhiều nhất và có thứ tự từ điển nhỏ nhất Ví dụ

Input	Output
1 abcdzzzzu abcd	 z

Bài 6. Kiểm tra xâu pangram bằng mảng đánh dấu và set.

Xâu được gọi là pangram nếu nó có đủ các chữ cái từ a tới z không phân biệt hoa thường, kiểm tra một xâu nhập vào có phải là xâu pangram hay không?

Input

Dòng đầu tiên là số lượng test case T . (1≤T≤100).

Mỗi test case bao gồm một xâu chỉ gồm các kí tự là chữ cái không quá 10000 kí tự

Output

In YES nếu xâu nhập vào là xâu pangram, ngược lại in NO

Ví dụ

Input	Output
2 ThequickbrownfoxjumpsoverthelazyDOG andrewneiMan	 YES NO

Bài 7. Đếm số lượng từ khác nhau trong câu

Input

Dòng đầu tiên là số lượng test case T . (1≤T≤100).

Mỗi dòng là một xâu bao gồm cả khoảng trắng có không quá 10000 kí tự

Output

In ra số lượng từ khác nhau trong câu

Ví dụ

Input	Output
2 Python C++ java php Python python Andrew neiman heisenberg neiman	 5 3

Bài 8.

Tìm từ có số lần xuất hiện nhiều nhất trong chuỗi, trường hợp các từ có số lần xuất hiện giống nhau từ có thứ tự từ điển nhỏ hơn sẽ là kết quả.

Input

Dòng đầu tiên là số lượng test case T . (1≤T≤100).

Mỗi dòng là một xâu bao gồm cả khoảng trắng có không quá 10000 kí tự

Output

In ra từ có số lần xuất hiện nhiều nhất và có thứ tự từ điển nhỏ nhất Ví dụ

Input	Output
1 ngon ngu lap trinh ngon ngu	ngon

Bài 9. Từ lặp đầu tiên

Tìm từ được lặp lại đầu tiên trong câu. Input
Dòng đầu tiên là số lượng bộ test (1 ≤T≤100).
T dòng tiếp theo mỗi dòng chứa một chuỗi đầu vào. Output
Từ đầu tiên được lặp lại, dữ liệu đảm bảo câu có 2 từ trở lên vào có xuất hiện từ được lặp lại.

Input	Output
2 abc abc abc zzz zzz cd ngon ngu lap lap ngu ngon	abc lap

Bài 10. Đặt tên người dùng

Xây dựng chương trình đặt tên tài khoản người dùng. Nếu tên người dùng muốn đặt đã xuất hiện trong hệ thống thì sẽ đặt tên tài khoản theo cú pháp "tên người dùng muốn đặt" +số tài khoản cùng tên trong hệ thống cộng thêm 1.
Ví dụ: Giả sử trong hệ thống đã tồn tại tên người dùng rech thì người dùng tiếp theo muốn sử dụng tên tài khoản là rech sẽ được lưu ở hệ thống với tên rech1, tương tự như vậy trong trường hợp có 2 tài khoản tên rech trong hệ thống thì người dùng có tên rech sẽ được lưu với tên rech2. Input
Dòng đầu tiên là n số lượng tên người dùng muốn cài đặt vào hệ thống, n dòng tiếp theo sẽ là tên người dùng, tên người dùng chỉ bao gồm 1 từ duy nhất Ouput
In kết quả là tên người dùng được lưu trong hệ thống.

Input	Output

14 an binh an binh long huong ngoc thuan nhung nhung ngoc thuan nhung nhung nhung	 an binh an1 binh1 long huong ngoc thuan nhung nhung1 ngoc thuan nhung2 nhung3 nhung4

VECTOR

Bài 1 : Viết hàm khai báo và trả về vector gồm n phần tử với các phần tử có giá trị từ 1 tới n.

Bài 2 : Cho một vector kiểu số nguyên v là đầu vào của hàm, hãy trả về tổng các phần tử trong vector này.

Ví dụ:

Với v = [1, 3, 2, 4] thì sumOfVectorElements(v) = 10.
Giải thích: 1 + 3 + 2 + 4 = 10.
Với v = [1, -4, 10] thì sumOfVectorElements(v) = 7.

Bài 3 :

Cho một vector chứa các số nguyên, bạn hãy viết hàm trả về tổng của các phần tử lẻ trong vector đó.

Ví dụ:

Với v = [1, 3, 2, 4] thì sumOfOddElements(v) = 4.

Giải thích: 1 + 3 = 4.
Với v = [1, -4, 6, 7, 8, -3] thì sumOfOddElements(v) = 5.
Giải thích: 1 + 7 + (-3) = 5.

Bài 4 : Cho trước một vector và một số tự nhiên n, hãy thay đổi kích thước của vector đó về n.

5 3	1 2 3 4 5
3 5	1 2 3 0 0

Bài 5 :
Cho một vector các số nguyên v, hãy sắp xếp các số nguyên trong vector này theo thứ tự tăng dần.

Ví dụ

Với v = [4, 5, 3, 2], thì sortVector(v) = [2, 3, 4, 5].
Với v = [3, 2, 1], thì sortVector(v) = [1, 2, 3].

Bài 6 :
Cho một vector chứa các số nguyên, hãy tìm giá trị lớn nhất của tích của 2 số nguyên liên tiếp trong vector (đầu vào luôn đảm bảo vector có ít nhất 2 phần tử).

Ví dụ:

Với v = [3, 5, 2, 9, 1], thì maxProductOfAdjacentElements(v) = 18.
Giải thích: Tích 2 số liên tiếp lớn nhất trong v là v[2] * v[3] = 18.
Với v = [1, 2, 3, 1], thì maxProductOfAdjacentElements(v) = 6.

Bài 7 : Cho một vector chứa các xâu kí tự, bạn hãy viết hàm tìm những xâu có độ dài lớn nhất trong vector ban đầu và trả về kết quả tương ứng.

Ví dụ:

Với đầu vào arr = ["aba", "aa", "ad", "vcd", "aba"], kết quả là findLongestStrings(arr) = ["aba", "vcd", "aba"].

Bài 8 : Cho trước một vector các số nguyên v, bạn hãy viết hàm xóa phần tử cuối cùng trong vector này và trả về vector tương ứng sau khi xóa.

Ví dụ:

Với v = [1, 2, 3, 4], thì removeLastElement(v) = [1, 2, 3].
Với v = [1, 2, 3], thì removeLastElement(v) = [1, 2]

Bài 9 : Cho vector gồm các số nguyên v. Bạn hãy viết hàm trả về tổng của phần tử đầu tiên và cuối cùng trong vector đó.

Ví dụ:

Với v = [1, 2, 4, 9] thì sumOfFirstAndLastElement(v) = 10.

Giải thích: 1 + 9 = 10.

Với v = [1, 3, 4] thì sumOfFirstAndLastElement(v) = 5.

Bài 10 :

Cho trước vector v và hai số nguyên l, r. Bạn hãy viết hàm xóa đi các phần tử có chỉ số từ l tới r trong vector v và trả về vector kết quả tương ứng.

Ví dụ:

Với v = [1, 2, 3, 4, 5, 6, 7], l = 1, r = 3 thì
removeElements(v) = [1, 5, 6, 7].
Với v = [2, 5, 8, 9, 6], l = 3, r = 3 thì removeElements(v) = [2, 5, 8, 6].

Set ::
Bài 1 : Cho một vector chứa các số nguyên.

Hãy đưa ra số lượng phần tử khác nhau trong vector đó.

Ví dụ:

Với inputVector = [1, 3, 3, 2],
thì differentNumbers(inputVector) = 3.
Giải thích: Có 3 phần tử khác nhau trong vector là: 1, 3, 2

Với inputVector = [3, 3, 3],
thì differentNumbers(inputVector) = 1.

Bài 2 : Cho một ma trận gồm các dãy nhị phân khác nhau.

Hãy đưa ra các dãy nhị phân khác nhau trong ma trận đó.

Ví dụ:

Với matrix =

[[1,1,0,1],

[1,0,0,1],

[1,1,0,1]].

thì đầu ra sẽ là: uniqueRow(matrix) =

[[1,1,0,1],

[1,0,0,1]].

Bài 3 :

Anh Việt đang thông kê số liệu cho công ty, anh muốn giá trị nhỏ thứ hai ở trong một dãy số nguyên.

Hãy giúp anh Việt tìm ra giá trị đó, nếu không có kết quả như yêu cầu thì trả về "NO".

Ví dụ:

Với arr = [1, 2, 3, 1, 1] thì kết quả sẽ là secondOrder(arr) = "2".

Với arr = [-4, 1, 2, 2] thì kết quả sẽ là secondOrder(arr) = "1".

Bài 4 :
Cho một dãy gồm các số nguyên và một số nguyên k.

Hãy kiểm tra xem trong dãy đó có phần tử giá trị k hay không, trả về true nếu có, false nếu không.

Ví dụ:

Với arr = [1, 2, 5, 3] và k = 5 thì setFind(arr, k) = true.

Với arr = [4, 6, 2, 7] và k = 3 thì setFind(arr, k) = false.

Bài 5 :
Cho hai dãy arr1 và arr2 thuộc kiểu vetor<string>. Bạn tạo một dãy từ hai dãy trên theo điều kiện sau:

Kết quả chỉ chứa các chuôi riêng biệt (Không có hai chuỗi nào giống nhau).
Kết quả chứa các phần tử trong arr1 mà không xuất hiện trong arr2.
Các phần tử các dãy kết quả được sắp xếp theo thứ tự từ điểm từ nhỏ đến lớn.
Ví dụ:

Với arr1 = ["codelearn", "learncode", "codelearn", "io","fpt"]
arr2 = ["learncode", "codelearnio", "fsoft"].
Thì kết quả sẽ là mergeStringArr(arr1, arr2) = ["codelearn", "fpt", "io"]

Bài 6 :
Cho một dãy số nguyên arr và số nguyên k.

Hãy tìm ra hai số:

Số m là số nhỏ nhất trong dãy lớn hơn k. Nếu không có thì m = -1.
Số n là số nhỏ nhất trong dãy lớn hơn hoặc bằng k. Nếu không có thì n = -1.

Kết quả trả về là dãy gồm 2 số [m,n].

Ví dụ:

Với arr = [1, 2, 3, 4, 5] và k = 4 thì setFunction(a, k) = [5,4].
Với arr = [1, 2, 3] và k = 3 thì setFunction(a, k) = [-1,3]

Bài 7 :
Một vector được gọi là vector beautifull nếu một số trong vector đó chỉ xuất hiện đúng một lần.

Cho vector v gồm các số nguyên. Cần xóa ít nhất bao nhiêu phần tử trong v để vector v trở thành vector beautifull .

Ví dụ:

Với v = [2, 3, 6, 3] thì vectorBeautifull(v) = 1.
Giải thích: Cần xóa một số 3.

Với v = [1, 2, 3] thì vectorBeautifull(v) = 0.

Bài 8 :
Việt đang muốn tạo ra những từ khác nhau từ chuỗi word. Với một chu kỳ, Anh bắt đầu bằng cách lấy ký tự cuối cùng của chuỗi word và di chuyển nó lên đầu. Anh có thể thực hiện chu kỳ này rất nhiều lần.

Ví dụ lên tục thực hiện chu kỳ biến đổi đó trên word = "abcda" thì anh Việt sẽ nhận được các từ "aabcd", "daabc",...

Anh Việt muốn biết rằng mình có thể tạo được bao nhiêu từ riêng biệt bằng cách biến đổi trên.

Ví dụ:

Với word = "abcd" thì kết quả sẽ là cyclicWord(word) = 4
các từ mà anh Việt có thể nhận được là "abcd", "dabc", "cdab" và "bcda".

Bài 9 :
Tính số lượng tiểu thiểu các ký tự cần thay đổi trong chuỗi str để chuỗi đó có k ký tự khác nhau hoặc xuất ra điều đó là không thể.

Chuỗi str chỉ bao gồm những chữ cái latinh viết thường và cũng chỉ có thể thay thành những chữ cái latinh viết thường.

Ví dụ:

Với str = "yandex", k = 6 thì kết quả sẽ là diversity(s, k) = "0"

Với str = "google", k = 7 thì kết quả sẽ là diversity(s, k) = "impossible".

Với str = "codelearn", k = 9 thì kết quả sẽ là diversity(s, k) = "1"

Bài 10 :
Trong ngôn ngữ Aramic từ có thể đại diện cho các đối tượng.

Các từ trong Aramic có tính chất đặc biệt:

Một từ là một gốc nếu nó không chứa cùng một chữ cái nhiều lần.
Một gốc và tất cả các hoán vị của nó cũng chỉ đại diện cho cùng một đối tượng.
Từ gốc x của một từ y là từ chứa tất cả các chữ cái xuất hiện trong y theo cách mà mỗi chữ cái xuất hiện một lần. Ví dụ: gốc của "aaaa", "aa", "aaa" là "a", gốc của "aabb", "bab", "baabb", "ab" là "ab".
Bất kỳ từ nào trong Aramic đại diện cho cùng một đối tượng với gốc của nó.
Bạn có một dãy từ Aramic. Hãy đưa ra số lượng đối tượng khác nhau trong dãy đó.

Ví dụ:

Với words = ["a","aa","aaa","ab","abb"] thì kết quả là: aramic(words) = 2.

Bài 11 :
An có một danh sách các sản phẩm trong siêu thị nhưng thật không may trong danh sách lại có những sản phẩm xuất hiện nhiều lần. Bạn hãy giúp An làm lại danh sách sao cho một sản phẩm chỉ xuất hiện một lần trong danh sách và các sản phẩm xuất hiện theo thứ tăng dần trong từ điển.

Ví dụ:

Với products = ["watermelon", "grapes", "grapes", "apple", "grapes"]
thì getDistinctProducts(products) = ["apple", "grapes", "watermelon"].

Mappp
Bài 1 :
Cho một chuỗi s, hãy đưa ra một dãy lần lượt là các ký tự và số lần xuất hiện của nó, các ký tự sắp xếp theo thự tự từ điển.

Ví dụ:

Với s = "aacccd" thì countChar = ["a 2", "c 3", "d 1"].

Với s = "aabbbca" thì countChar = ["a 3", "b 3", "c 1"].

Bài 2 : Bạn đã được cung cấp một chuỗi s làm đầu vào và bạn phải in biểu mẫu đã sửa đổi. Chuỗi được sửa đổi theo cách sau:

Những ký tự giống nhau chỉ lấy duy nhất một ký tự.
Ký tự nào có tần số xuất hiện trong chuỗi s nhiều hơn được sắp xếp trước.
Những ký tữ có cùng tần số xuất hiện thì sắp xếp theo thứ từ từ điển từ nhỏ đến lớn.
Ví dụ:

Với s = "codelearn" thì modifyString = "eacdlnor".

Với s = "helloworld" thì modifyString = "lodehrw".

Bài 3 :
Cho 2 mảng các chuỗi arr1, arr2 dưới dạng vector. Ứng với mỗi chuỗi trong arr2 bạn cần tìm một chuỗi có thứ tự từ điển nhỏ nhất trong arr1 mà đứng sau arr2 trong từ điển. Nếu không có chuỗi nào trong arr1 thỏa mãn thì output "-1".

Ví dụ

Với arr1 = ["codelearn","learncode", "io"],
arr2 = ["code","war","io"]
Thì kết quả mong muốn greaterString(arr1, arr2) = ["codelearn","-1","learncode"]

Bài 4 :
Cho hãy dãy số nguyên arr1 và arr2, hãy tính tổng những số xuất hiện trong cả hai dãy, lưu ý là mỗi số chỉ được tính một lần.

Ví dụ:

Với arr1 = [6, 7, 5, 4, 6, 8], arr2 = [2, 5, 7, 5, 3] thì sumOfCommon(arr1, arr2) = 12.

Với arr1 = [5,6,7], arr2 = [2,3,4] thì sumOfCommon(arr1, arr2) = 0.

Bài 5 :
Cho một mảng các chuỗi chuỗi chữ thường với các phần tử có thể trùng lặp. Hãy đưa ra khoảng cách lớn nhất giữa hai phần tử giống nhau ở trong dãy đó.

Nếu không có bất kỳ hai chuỗi nào giống nhau thì trả về 0.

Ví dụ:

Với arr = ["codelearn", "io", "programmer", "codelearn", "programmer"]. thì maximumDifference(arr) = 3.
Giải thích: hai phần tử giống nhau có khoảng cách lớn nhất trong trường hợp trên là arr[0] và arr[3].

Bài 6 :
Cho mỗi dãy số nguyên và một dãy số nguyên arr và một số nguyên dương sum.

Hãy kiểm tra xem dãy số có tồn tại hai số có tổng bằng sum hay không.

Ví dụ:

Với arr = [2,4,-1,9,8], sum = 6 thì checkSum(arr, sum) = true.

Với arr = [2,5,3,8,9], sum = 3 thì checkSum(arr, sum) = false.

Với arr = [4,7,3,5], sum = 6 thì checkSum(arr, sum) = false.

Bài 7

Cho một dãy gồm các số nguyên, hãy đơn ra số thỏa mãn là số lớn nhất trong những số có tần số bé nhất.

Ví dụ:

Với arr = [2,2,4,4,7,7,7] thì largestElement(arr) = 4.

Với arr = [1,3,4,5,5] thì largestElement(arr) = 4.

Bài 8 :
Cho một danh bạ điện thoại và danh sách các tên. Ứng với mỗi tên trong danh sách hãy đếm xem tên này là bắt đầu của bao nhiêu tên trong danh bạ điện thoại, hay nói cách khác hãy đếm xem tên này là prefix của bao nhiêu tên trong danh bạ.

Ví dụ:

Với contacts = ["Codelearn", "Codewar"],
names = ["Code", "Codel", "io"]
thì countPrefix(contacts, names) = [2,1,0].

Bài 10 : Cho danh sách các sản phẩm của 2 kho hàng A và B. Do chiến lược kinh doanh bạn được giao nhiệm nhập các sản phẩm từ kho B vào kho A sao cho những sản phẩm nào đã có trong kho A thì không nhập.

Ví dụ:

Với A = ["Banana", "Banana", "Apple"],
B = ["Orange", "Apple", "Orange", "Watermelon"],
thì mergeProducts(A, B) = [true, false, false, true].
Lưu ý: Sản phẩm thứ 3 trong kho B là "Orange" sẽ không được nhập vì trong kho A đã có một sản phẩm như vậy.