

LUYỆN TẬP LẬP TRÌNH

CONTEST 4

BÀI A: HÀNG ĐỢI

Yêu cầu bạn xây dựng một queue với các truy vấn sau đây:

“PUSH x”: Thêm phần tử x vào cuối của queue ($0 \leq x \leq 1000$).

“PRINTFRONT”: In ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

“POP”: Xóa phần tử ở đầu của queue. Nếu queue rỗng, không làm gì cả.

Input:

Dòng đầu tiên là số lượng truy vấn Q ($Q \leq 100000$).

Mỗi truy vấn có dạng như trên.

Output:

Với mỗi truy vấn “PRINT”, hãy in ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

Test ví dụ:

Input:	Output
9	2
PUSH 1	2
PUSH 2	NONE
POP	
PRINTFRONT	
PUSH 3	
PRINTFRONT	
POP	
POP	
PRINTFRONT	

BÀI B: ĐIỂM XUẤT PHÁT

Có N thành phố liên kết với nhau tạo thành một vòng tròn và các đường đi của chúng chỉ là một chiều, khoảng cách từ thành phố thứ i tới i+1 (trạm đến tiếp theo) bằng $D[i]$. Bạn có một chiếc xe tải có khả năng chứa rất nhiều (vô hạn xăng), tuy nhiên lại rất tốn xăng, tận 1 lít xăng / 1 đơn vị chiều dài. Tại mỗi thành phố có một trạm xăng có lượng xăng bằng $A[i]$ lít.

Mục tiêu của bạn là có một hành trình đi hết N thành phố rồi quay lại đúng điểm xuất phát. Tuy nhiên, việc lựa chọn điểm xuất phát là không hề dễ dàng, vì nếu không tối ưu, có thể bạn sẽ bị hết xăng khi đi giữa đường.

Nhiệm vụ của bạn là xác định xem có thể tìm được điểm xuất phát sao cho có thể đi trọn vẹn một vòng quanh các thành phố hay không?

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test gồm một số nguyên N ($N \leq 100\,000$).

N dòng tiếp theo, mỗi dòng gồm 2 số nguyên A[i] và D[i] ($1 \leq A[i], D[i] \leq 10^6$).

Output:

Với mỗi test, in ra “YES” nếu có thể đi một vòng quanh N thành phố, in ra “NO” trong trường hợp ngược lại.

Test ví dụ:

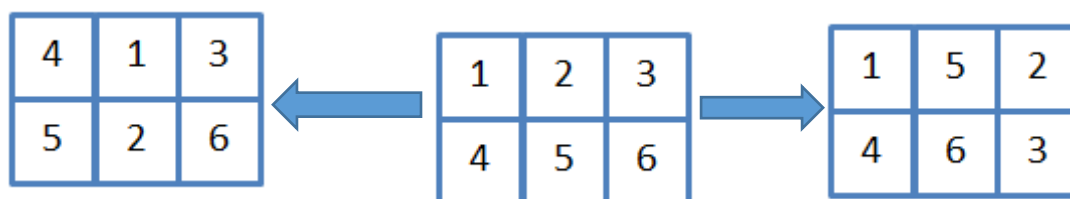
Input:	Output
3	YES
3	YES
6 4	NO
3 6	
7 3	
4	
5 5	
5 5	
5 5	
5 5	
2	
1 2	
2 3	

Giải thích test 1: Xuất phát tại thành phố thứ 3.

BÀI C: QUAY HÌNH VUÔNG

Time limit: 1s

Có một chiếc bảng hình chữ nhật với 6 miếng ghép, trên mỗi miếng ghép được điền một số nguyên trong khoảng từ 1 đến 6. Tại mỗi bước, chọn một hình vuông (bên trái hoặc bên phải), rồi quay theo chiều kim đồng hồ.



Yêu cầu: Cho một trạng thái của bảng, hãy tính số phép biến đổi ít nhất để đưa bảng đến trạng thái đích.

Input:

Dòng đầu tiên chứa 6 số là trạng thái bảng ban đầu (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

Dòng thứ hai chứa 6 số là trạng thái bảng đích (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

Output:

In ra một số nguyên là đáp số của bài toán.

Test ví dụ:

Input	Output
1 2 3 4 5 6 4 1 2 6 5 3	2

BÀI D: DI CHUYỂN

Cho một bảng kích thước $N \times N$, trong đó có các ô trống '.' và vật cản 'X'. Các hàng và các cột được đánh số từ 0.

Mỗi bước di chuyển, bạn có thể đi từ ô (x, y) tới ô (u, v) nếu như 2 ô này nằm trên cùng một hàng hoặc một cột, và không có vật cản nào ở giữa.

Cho điểm xuất phát và điểm đích. Bạn hãy tính số bước di chuyển ít nhất?

Input:

Dòng đầu tiên là số nguyên dương N ($1 \leq N \leq 100$).

N dòng tiếp theo, mỗi dòng gồm N kí tự mô tả bảng.

Cuối cùng là 4 số nguyên a, b, c, d với (a, b) là tọa độ điểm xuất phát, (c, d) là tọa độ đích. Dữ liệu đảm bảo hai vị trí này không phải là ô cấm.

Output:

In ra một số nguyên là đáp số của bài toán.

Test ví dụ:

Input	Output
3 .X. .X. ... 0 0 0 2	3

BÀI E: BỘI SỐ LỚN NHẤT CỦA 3

Cho dãy số $A[]$ có N phần tử là các chữ số từ 0 đến 9. Nhiệm vụ của bạn là hãy chọn lấy một tổ hợp các phần tử và sắp xếp chúng sao cho thu được số lớn nhất chia hết cho 3.

Nếu không tìm được số nào, in ra -1.

Input:

Dòng đầu tiên là số lượng bộ test T ($1 \leq T \leq 50$).

Mỗi test bắt đầu bởi số nguyên N ($1 \leq N \leq 100\,000$).

Dòng tiếp theo gồm N số nguyên $A[i]$ ($0 \leq A[i] \leq 9$).

Output:

Với mỗi test, in ra đáp án tìm được trên một dòng.

Test ví dụ:

Input	Output
3	981
3	8760
8 1 9	-1
5	
8 1 7 6 0	
2	
5 2	

BÀI F: Dãy con tăng dài nhất

Cho dãy số $A[]$ có N phần tử. Bạn hãy tìm dãy con tăng dài nhất (dãy tăng chặt).

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test gồm số nguyên dương N .

Dòng tiếp theo gồm N số nguyên $A[i]$ ($0 \leq A[i] \leq 100\,000$).

Output:

Với mỗi test, in ra độ dài dãy con tăng dài nhất trên một dòng.

Subtask 1: $N \leq 1000$.

Subtask 2: $N \leq 100\,000$.

Test ví dụ:

Input:	Output
2	4
5	6
1 2 3 4 4	
9	
2 5 3 7 11 8 10 13 6	

BÀI G: ĐƯỜNG ĐI NHỎ NHẤT

Cho bảng $A[]$ kích thước $N \times M$ (N hàng, M cột). Bạn được phép đi sang trái, đi sang phải và đi xuống ô chéo dưới. Khi đi qua ô (i, j) , điểm nhận được bằng $A[i][j]$.

Hãy tìm đường đi từ ô $(1, 1)$ tới ô (N, M) sao cho tổng điểm là nhỏ nhất.

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test gồm số nguyên dương N và M .

N dòng tiếp theo, mỗi dòng gồm M số nguyên $A[i][j]$ ($0 \leq A[i] \leq 1000$).

Output:

Với mỗi test, in ra độ dài dãy con tăng dài nhất trên một dòng.

Test ví dụ:

Input:	Output
1 3 3 1 2 3 4 8 2 1 5 3	8

Giải thích test: Đường đi $(1, 1) \rightarrow (1, 2) \rightarrow (2, 3) \rightarrow (3, 3)$.

BÀI H: DÂY CON CHUNG DÀI NHẤT

Time limit: 1s

Cho 2 chuỗi S và T . Hãy tìm dãy con chung dài nhất của 2 chuỗi này (các phần tử không nhất thiết phải liên tiếp nhau).

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test gồm hai dòng, mô tả chuỗi S và T , mỗi chuỗi có độ dài không quá 1000 và chỉ gồm các chữ cái in hoa.

Output:

Với mỗi test, in ra độ dài dãy con chung dài nhất trên một dòng.

Test ví dụ:

Input:	Output
2 AGGTAB GXTXAYB AA BB	4 0

Giải thích test 1: Dãy con chung là G, T, A, B.

BÀI I: TỔ HỢP

Cho 2 số nguyên n, k . Bạn hãy tính $C(n, k)$ modulo 10^9+7 .

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test gồm 2 số nguyên n, k ($1 \leq k \leq n \leq 1000$).

Output:

Với mỗi test, in ra đáp án trên một dòng.

Test ví dụ:

Input:	Output
2	10
5 2	120
10 3	

BÀI J: XÂU CON ĐỐI XỨNG DÀI NHẤT

Cho chuỗi S . Hãy tìm chuỗi con đối xứng dài nhất của S .

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 10$).

Mỗi test gồm một chuỗi S có độ dài không vượt quá 5000, chỉ gồm các ký tự thường.

Output:

Với mỗi test, in ra đáp án tìm được.

Test ví dụ:

Input:	Output
2	5
abcbadd	5
aaaaa	

BÀI K: SỐ CÓ CÁC CHỮ SỐ SẮP XẾP TĂNG DẦN

Xét các số X có các chữ số được sắp xếp tăng dần, tức $a[1] \leq a[2] \leq \dots \leq a[N]$ trong đó $a[1], a[2], \dots, a[N]$ lần lượt là các chữ số của X . Chẳng hạn 223, 8999, ...

Nhiệm vụ của bạn là đếm xem có bao nhiêu số như vậy với mỗi N cho trước.

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 100$).

Mỗi test gồm một số nguyên dương N ($1 \leq N \leq 100$).

Output:

Với mỗi test, in ra đáp án tìm được trên một dòng theo modulo 10^9+7 .

Test ví dụ:

Input:	Output
3	10
1	55
2	220
3	

BÀI L: BẬC THANG

Một chiếc cầu thang có N bậc. Mỗi bước, bạn được phép bước lên trên tối đa K bước. Hỏi có tất cả bao nhiêu cách bước để đi hết cầu thang? (Tổng số bước đúng bằng N).

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 100$).

Mỗi test gồm hai số nguyên dương N và K ($1 \leq N \leq 100000$, $1 \leq K \leq 100$).

Output:

Với mỗi test, in ra đáp án tìm được trên một dòng theo modulo 10^9+7 .

Test ví dụ:

Input:	Output
2	2
2 2	5
4 2	

Giải thích test 1: Có 2 cách đó là (1, 1) và (2).

Giải thích test 2: 5 cách đó là: (1, 1, 1, 1), (1, 1, 2), (1, 2, 1), (2, 1, 1), (2, 2).

BÀI M: HÀNH TRÌNH DU LỊCH

Trong kì nghỉ hè năm nay Tí được bố thưởng cho 1 tour du lịch quanh N đất nước tươi đẹp với nhiều thắng cảnh nổi tiếng. Tất nhiên Tí sẽ đi bằng máy bay.

Giá vé máy bay từ đất nước i đến đất nước j là $C[i][j]$ (đĩ nhiên $C[i][j]$ có thể khác $C[j][i]$). Tuy được bố thưởng cho nhiều tiền để đi du lịch nhưng Tí cũng muốn tìm cho mình 1 hành trình với chi phí rẻ nhất có thể để dành tiền mua quà về tặng mọi người.

Bạn hãy giúp Tí tìm 1 hành trình đi qua tất cả các nước, mỗi nước đúng 1 lần sao cho chi phí là bé nhất nhé.

Input:

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$).

Mỗi test bắt đầu bởi số nguyên dương N ($5 \leq N \leq 15$).

N dòng tiếp theo, mỗi dòng gồm N số nguyên mô tả chi phí $C[i][j]$ ($1 \leq C[i][j] \leq 10000$).

Output:

Với mỗi test, in ra đáp án tìm được trên một dòng.

Test ví dụ:

Input:	Output
1 6 0 1 2 1 3 4 5 0 3 2 3 4 4 1 0 2 1 2 4 2 5 0 4 3 2 5 3 5 0 2 5 4 3 3 1 0	8

Giải thích test: 1 hành trình tối ưu là $3 \rightarrow 6 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 4$.

BÀI N: CHUYỂN N THÀNH 1

Cho số nguyên dương N ($N < 2^{31}$). Chỉ được phép sử dụng hai thao tác A, B dưới đây, hãy dịch chuyển N về 1 sao cho số các thao tác A, B được thực hiện ít nhất.

Thao tác A: Biến đổi $N = N - 1$.

Thao tác B: Biến đổi $N = \max(u, v)$, trong đó $u * v = N$ ($u > 1, v > 1$).

Ví dụ $N = 17$ được thực hiện ít nhất 4 bước A, B như sau:

Thao tác A: $N = N - 1 = 17 - 1 = 16$

Thao tác B: $N = \max(u, v) = \max(4 * 4) = 4$

Thao tác B: $N = \max(u, v) = \max(2 * 2) = 2$

Thao tác A: $N = N - 1 = 2 - 1 = 1$

Dữ liệu vào:

- Dòng đầu tiên ghi lại T là số lượng bộ test ($T \leq 10$).
- Mỗi bộ test gồm một dòng ghi lại một số nguyên không âm.

Kết quả:

- Ứng với mỗi bộ test đưa ra số các bước A, B thực hiện ít nhất để dịch chuyển test tương ứng về 1.

Ví dụ:

Input	Output
6	4

17	5
33	5
255	5
1024	6
1029	6
65535	

BÀI 0: XÂU ĐỐI XỨNG

Một chuỗi ký tự được gọi là chuỗi đối xứng nếu đọc chuỗi đó từ trái qua phải cũng như đọc nó từ phải qua trái ta thu được cùng một chuỗi. Một chuỗi ký tự được gọi là chuỗi đối xứng cấp 0 nếu tồn tại một cách sắp xếp lại các ký tự của nó để thu được một chuỗi đối xứng. Một chuỗi $s = s_1s_2 \dots s_n$ gồm n ký tự (ta gọi n là độ dài của chuỗi s) được gọi là chuỗi đối xứng cấp k nếu nó thỏa mãn các điều kiện sau:

- 1) s là chuỗi đối xứng cấp 0;
- 2) tồn tại k vị trí $1 < i_1 < i_2 < \dots < i_k < n$, sao cho chuỗi con gồm i_j ký tự đầu tiên của chuỗi s là chuỗi đối xứng cấp 0, $j = 1, 2, \dots, k$.

Dễ thấy, nếu một chuỗi là chuỗi đối xứng cấp k thì nó cũng là chuỗi đối xứng cấp m với $0 \leq m < k$.

Ví dụ, các chuỗi ‘ada’, ‘abba’ là các chuỗi đối xứng; chuỗi ‘daa’ là chuỗi đối xứng cấp 0; chuỗi ‘abab’ là chuỗi đối xứng cấp 1 (vị trí thỏa mãn định nghĩa là $i_1 = 3$); chuỗi ‘ababd’ là chuỗi đối xứng cấp 2 (2 vị trí thỏa mãn định nghĩa là $i_1 = 3$ và $i_2 = 4$).

Ký hiệu $S(n, k, \Omega)$ là dãy tất cả các chuỗi đối xứng cấp k độ dài n chỉ gồm các ký tự thuộc tập ký tự Ω được liệt kê theo thứ tự từ điển, đánh số bắt đầu từ 1.

Ví dụ, với $n=3$; $k=1$; $\Omega=\{‘v’, ‘n’\}$, dãy $S(3, 1, \{‘v’, ‘n’\})$ gồm 4 chuỗi được liệt kê theo thứ tự từ điển và đánh số thứ tự bắt đầu từ 1 sau đây:

1. ‘nnn’
2. ‘nnv’
3. ‘vvn’
4. ‘vvv’

Yêu cầu: Cho n, k, Ω và số nguyên t , hãy đưa ra chuỗi thứ t trong dãy $S(n, k, \Omega)$.

Dữ liệu: Dòng đầu chứa hai số nguyên không âm n, k ($k \leq n-2$; $2 \leq n \leq 50$);

- Dòng thứ hai chứa các ký tự của tập Ω được ghi nhận bởi một chuỗi gồm không quá 5 chữ cái in thường lấy từ tập 26 chữ cái tiếng Anh từ ‘a’ đến ‘z’;
- Dòng thứ ba chứa số nguyên dương t (t không lớn hơn số lượng phần tử của $S(n, k, \Omega)$).

Các số trên cùng dòng được ghi cách nhau bởi dấu cách.

Kết quả: Ghi ra chuỗi thứ t của dãy $S(n, k, \Omega)$.

Ví dụ:

Input	Output
3 1 vn 2	nnv

BÀI P: SỐ ĐẶC BIỆT HOÀN HẢO

Số nguyên dương N được gọi là số đặc biệt hoàn hảo nếu N thỏa mãn

$$\delta^2(N) = \delta(\delta(N)) = 2*N$$

Trong đó, $\delta(N)$ là tổng các ước số thực sự của N kể cả N . Ví dụ $N = 16$ là số đặc biệt hoàn hảo vì

$$\delta(16) = 1 + 2 + 4 + 8 + 16 = 31$$

$$\delta(\delta(16)) = \delta(31) = 1 + 31 = 32 = 2*16.$$

Số $N = 8$ không phải số đặc biệt hoàn hảo vì

$$\delta(8) = 1 + 2 + 4 + 8 = 15$$

$$\delta(\delta(8)) = \delta(15) = 1 + 3 + 5 + 15 = 24 \neq 2*8.$$

Cho số tự nhiên N ($1 \leq N \leq 2^{31}$). Hãy kiểm tra xem N có phải là số đặc biệt hoàn hảo hay không.

Dữ liệu vào:

Dòng đầu tiên là số lượng bộ test T ($T \leq 50$).

Mỗi test gồm một dòng ghi lại một số nguyên không âm.

Kết quả:

Ứng với mỗi test in ra “YES” nếu số đã cho là số đặc biệt hoàn hảo. In ra “NO” nếu số đã cho không phải là số đặc biệt hoàn hảo.

Ví dụ:

Input	Output
6	YES
4	NO
8	YES
64	NO
128	NO
512	YES
65536	

BÀI Q: VIRUS

“TextFile” là một virus chuyên tấn công các file văn bản theo phương thức sau: Sao chép một đoạn các ký tự liên tiếp trong nội dung của file văn bản vào bộ nhớ trong, thay đổi một số ký tự trong đoạn này, sau đó chèn đoạn văn bản đã thay đổi vào ngay sau đoạn văn bản vừa sao chép trong file văn bản.

Vinh đang phát triển phần mềm để phát hiện một file văn bản đã bị nhiễm virus nói trên hay chưa. Vì thế, Vinh cần giải quyết bài toán sau: Cho chuỗi ký tự T và số nguyên không âm k , chuỗi con gồm các ký tự từ vị trí p đến vị trí q của chuỗi T được gọi là đoạn có khả năng bị virus sao chép mức k nếu nó sai khác với chuỗi con gồm các ký tự từ vị trí $q+1$ đến vị trí $q+(q-p+1)$ của chuỗi T ở không quá k vị trí.

Ví dụ, xét chuỗi $T = \text{'zabaaxy'}$ và $k = 1$. Đoạn văn bản ‘ab’ từ ký tự thứ 2 đến ký tự thứ 3 là đoạn văn bản độ dài 2 có khả năng bị virus sao chép mức 1 vì nó khác với đoạn văn bản ‘aa’ gồm các ký tự từ ký tự thứ 4 đến ký tự thứ 5 của chuỗi T ở 1 vị trí.

Yêu cầu: Cho chuỗi ký tự T và n số nguyên không âm k_1, k_2, \dots, k_n . Với mỗi giá trị k_i , hãy tìm độ dài đoạn dài nhất trong chuỗi T có khả năng bị virus sao chép mức k_i ($i = 1, 2, \dots, n$).

Dữ liệu vào:

- Dòng đầu chứa số nguyên dương n ($n \leq 10$);
- Dòng thứ hai chứa một chuỗi T gồm các chữ cái in thường lấy từ tập 26 chữ cái tiếng Anh từ ‘a’ đến ‘z’ (độ dài chuỗi T không vượt quá 300);
- Dòng thứ i trong số n dòng tiếp theo ghi số nguyên không âm k_i ($k_i \leq 10, i = 1, 2, \dots, n$).

Kết quả: Gồm n dòng, dòng thứ i ghi một số nguyên không âm là độ dài đoạn dài nhất có khả năng bị virus sao chép mức $k_i, i=1, 2, \dots, n$. Ghi 0 nếu không tìm được đoạn như vậy.

Ví dụ:

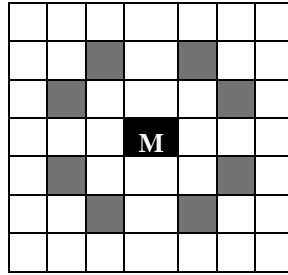
Input	Output
2	1
zabaaxy	2
0	
1	

Input	Output
2	2
zcaabcaaaa	4
0	
1	

Giải thích: Trong ví dụ bên phải, đoạn dài nhất có khả năng bị virus sao chép mức 0 là ‘aa’ có độ dài 2, đoạn dài nhất có khả năng bị virus sao chép mức 1 là ‘caab’ có độ dài 4.

BÀI R: QUÂN MÃ

Xét lưới ô vuông vô hạn trong đó có một số ô cấm, các ô còn lại là tự do. Các dòng và cột của lưới được đánh số theo thứ tự bởi các số nguyên $\dots -3 -2 -1 0 1 2 3 \dots$. Các cột được đánh số theo thứ tự từ trái sang phải, còn các dòng theo thứ tự từ dưới lên trên. Ô nằm trên giao của dòng x và cột y được gọi là ô (x, y) . Một quân mã đặt ở ô xuất phát là ô $(0,0)$. Sau một bước đi, ta có thể di quân mã đến một trong các ô ở đỉnh đối diện trên đường chéo của hình chữ nhật kích thước 2×3 .



Luật di chuyển của quân mã

Yêu cầu: Cho biết toạ độ của các ô cấm, vị trí ô đích nơi quân mã cần đến, hãy tìm cách di chuyển quân mã từ ô (0,0) đến ô đích sao cho số lượng bước đi cần thực hiện là ít nhất.

Dữ liệu vào: Dòng đầu tiên chứa T ($T \leq 3$) là số lượng test, tiếp đến là T nhóm dòng, mỗi nhóm chứa dữ liệu về một test theo khuôn dạng sau:

- Dòng đầu tiên chứa 2 số nguyên x_t, y_t được ghi cách nhau bởi dấu cách cho biết toạ độ của ô đích là (x_t, y_t) ;
- Dòng thứ hai chứa số nguyên dương n ($n \leq 1000$) là số lượng ô cấm;
- Dòng thứ i trong số n dòng tiếp theo chứa hai số nguyên được ghi cách nhau bởi dấu cách x_i, y_i cho biết (x_i, y_i) là toạ độ của ô cấm thứ i ($i = 1, 2, \dots, n$).

Chú ý : ($-10^3 \leq x_t, y_t \leq 10^3$; $-10^3 \leq x_i, y_i \leq 10^3$)

Kết quả: Gồm T dòng mỗi dòng chứa kết quả của một test tương ứng trong dữ liệu vào là số lượng bước đi ít nhất cần thực hiện để di chuyển quân mã từ ô xuất phát (0,0) đến ô đích. Ghi số -1 nếu như không thể di chuyển quân mã đến ô đích.

Ví dụ:

Input	Output
1 2 4 0	2