

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT – CONTEST 4

DANH SÁCH LIÊN KẾT – NGĂN XẾP VÀ ÁP DỤNG

A. NGĂN XẾP 1

Cho một ngăn xếp các số nguyên. Các thao tác được mô tả trong file văn bản gồm 3 lệnh: push, pop và show. Trong đó thao tác push kèm theo một giá trị cần thêm (không quá 1000). Hãy viết chương trình ghi ra kết quả của các lệnh show.

Dữ liệu vào: Gồm nhiều dòng, mỗi dòng chứa một lệnh push, pop hoặc show. Input đảm bảo số lượng phần tử trong stack khi nhiều nhất cũng không vượt quá 200.

Kết quả: Ghi ra màn hình các phần tử đang có trong stack theo thứ tự lưu trữ mỗi khi gặp lệnh show. Các số viết cách nhau đúng một khoảng trống. Nếu trong stack không còn gì thì in ra dòng “empty”

Ví dụ:

Input	Output
push 3	3 5
push 5	3 5 7
show	3
push 7	
show	
pop	
pop	
show	

B. NGĂN XẾP 2

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

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

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

“POP”: Xóa phần tử đầu tiên của stack. Nếu stack 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 stack. Nếu stack rỗng, in ra “NONE”.

Ví dụ:

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

C. BIỂU THỨC HẬU TỔ 1

Hãy sử dụng ngăn xếp để thực hiện việc chuyển các biểu thức trung tố sang hậu tố

Dữ liệu vào: Dòng 1 ghi số N (không quá 20) là số biểu thức trung tố (đúng khuôn dạng) chỉ bao gồm các phép cộng, trừ, nhân, chia, các chữ cái thường từ a đến z và các dấu ngoặc đơn. N dòng tiếp theo mỗi dòng ghi một biểu thức.

Kết quả: Ghi ra màn hình các biểu thức hậu tố kết quả.

Ví dụ:

Input	Output
1 ((a+b) * (c+d))	ab+cd+*

D. BIỂU THỨC HẬU TỔ 2

Cho một dãy biểu thức dạng trung tố có thể chưa đúng khuôn dạng với các phép toán cộng trừ nhân chia, lũy thừa.

Nhiệm vụ của bạn là hãy chuyển biểu thức đã cho về dạng hậu tố.

Chú ý xem ví dụ để hiểu sự khác biệt với bài trước.

Input:

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

Mỗi dòng gồm một xâu S, không quá 50 ký tự. Các số trong biểu thức đại diện bởi một chữ cái thường.

Output:

Với mỗi test, in ra biểu thức hậu tố tìm được.

Ví dụ:

Input:	Output
2 a+b*c+d a+b*(c^d-e)^(f+g*h)-i	abc*+d+ abcd^e-fgh*+^*+i-

E. DÂY NGOẶC ĐÚNG DÀI NHẤT

Cho một xâu chỉ gồm các ký tự '(' và ')'. Một dãy ngoặc đúng được định nghĩa như sau:

- Xâu rỗng là 1 dãy ngoặc đúng.
- Nếu A là 1 dãy ngoặc đúng thì (A) là 1 dãy ngoặc đúng.
- Nếu A và B là 2 dãy ngoặc đúng thì AB là 1 dãy ngoặc đúng.

Cho một xâu S. Nhiệm vụ của bạn là hãy tìm dãy ngoặc đúng dài nhất xuất hiện trong xâu đã cho.

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

Mỗi test gồm một xâu S có độ dài không vượt quá 10^5 ký tự.

Output: Với mỗi test in ra một số nguyên là độ dài dãy ngoặc đúng dài nhất tìm được.

Ví dụ:

Input:	Output
3 ((()) () () () (())))	2 4 6

F. KIỂM TRA DẪY NGOẶC ĐÚNG

Cho một xâu chỉ gồm các kí tự ‘(’, ‘)’, ‘[’, ‘]’, ‘{’, ‘}’. Một dãy ngoặc đúng được định nghĩa như sau:

- Xâu rỗng là 1 dãy ngoặc đúng.
- Nếu A là 1 dãy ngoặc đúng thì (A), [A], {A} là 1 dãy ngoặc đúng.
- Nếu A và B là 2 dãy ngoặc đúng thì AB là 1 dãy ngoặc đúng.

Cho một xâu S. Nhiệm vụ của bạn là xác định xâu S có là dãy ngoặc đúng hay không?

Input:

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

Mỗi test gồm 1 xâu S có độ dài không vượt quá 100 000.

Output:

Với mỗi test, in ra “YES” nếu như S là dãy ngoặc đúng, in ra “NO” trong trường hợp ngược lại.

Ví dụ:

Input:	Output
2	YES
[()] { } { [() ()] () }	NO
[()]	

G. SỬA LẠI DẪY NGOẶC

Cho một xâu chỉ gồm các kí tự ‘(’, ‘)’ và có độ dài chẵn. Hãy đếm số lượng dấu ngoặc cần phải đổi chiều ít nhất, sao cho xâu mới thu được là một dãy ngoặc đúng.

Input:

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

Mỗi test gồm 1 xâu S có độ dài không vượt quá 100 000, chỉ gồm dấu (và).

Output:

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

Ví dụ:

Input:	Output
4	2
)) ((2
((((1
(((())	3
) (()) (((

H. XÓA DẤU NGOẶC

Cho biểu thức toán học đúng, bạn cần tìm tất cả các biểu thức đúng có thể bằng cách xóa bỏ các cặp dấu ngoặc tương ứng với nhau từ biểu thức ban đầu.

Ví dụ: Cho biểu thức: $(2+(2*2)+2)$ Các biểu thức tìm được:

$$(2+2*2+2)$$

$$2+(2*2)+2$$

$$2+2*2+2$$

Các biểu thức $(2+2*2)+2$ và $2+(2*2)+2$ không được chấp nhận vì không xóa đi các cặp dấu ngoặc tương ứng với nhau

Dữ liệu vào: Một dòng chứa biểu thức gồm các số nguyên không âm, các dấu +, -, *, / và dấu ngoặc đơn.

Biểu thức không quá 200 kí tự, có chứa ít nhất 1 và không quá 10 cặp dấu ngoặc.

Kết quả: In ra tất cả các biểu thức khác nhau thỏa mãn đầu bài theo thứ tự từ điển

Ví dụ

Input	Output
$(1 + (2 * (3 + 4)))$	$(1 + (2 * 3 + 4))$ $(1 + 2 * (3 + 4))$ $(1 + 2 * 3 + 4)$ $1 + (2 * (3 + 4))$ $1 + (2 * 3 + 4)$ $1 + 2 * (3 + 4)$ $1 + 2 * 3 + 4$

I. TỔNG ĐA THỨC

Cho hai đa thức có bậc không quá 10000 (chỉ viết ra các phần tử có hệ số khác 0). Hãy sử dụng danh sách liên kết đơn để viết chương trình tính tổng hai đa thức đó.

Dữ liệu vào: Dòng đầu ghi số bộ test. Mỗi bộ test có hai dòng, mỗi dòng ghi một đa thức theo mẫu như trong ví dụ. Số phần tử của đa thức không quá 20.

Chú ý: Bậc của các hạng tử luôn theo thứ tự giảm dần, trong đa thức chỉ có phép cộng và luôn được viết đầy đủ hệ số + số mũ (kể cả mũ 0).

Kết quả: Ghi ra một dòng đa thức tổng tính được (theo mẫu như ví dụ)

Ví dụ:

Input	Output
1 $3x^8 + 7x^2 + 4x^0$ $11x^6 + 9x^2 + 2x^1 + 3x^0$	$3x^8 + 11x^6 + 16x^2 + 2x^1 + 7x^0$

J. TÍNH TOÁN GIÁ TRỊ BIỂU THỨC

Cho biểu thức S với các toán tử +, -, *, / và dấu ngoặc (). Các toán hạng là các số có giá trị không vượt quá 100. Hãy tính giá trị biểu thức S. Phép chia thực hiện với số nguyên, input đảm bảo số bị chia luôn khác 0, đáp số biểu thức có không quá 10 chữ số.

Input:

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

Mỗi dòng gồm một xâu S, không quá 100 ký tự. Các toán hạng là các số nguyên không âm.

Output:

Với mỗi test, in ra đáp án tìm được.

Ví dụ:

Input	Output
4 $6 * 3 + 2 - (6 - 4 / 2)$ $100 + 99 * 22$ $6 * ((4 * 3) + 5)$ $1 - 2$	16 2278 102 -1

K. PHẦN TỬ BÊN PHẢI ĐẦU TIÊN LỚN HƠN

Cho dãy số $A[]$ gồm N phần tử. Với mỗi $A[i]$, bạn cần tìm phần tử bên phải đầu tiên lớn hơn nó. Nếu không tồn tại, in ra -1 .

Input:

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

Mỗi test bắt đầu bởi số nguyên N ($1 \leq N \leq 100000$).

Dòng tiếp theo gồm N số nguyên $A[i]$ ($0 \leq A[i] \leq 10^9$).

Output:

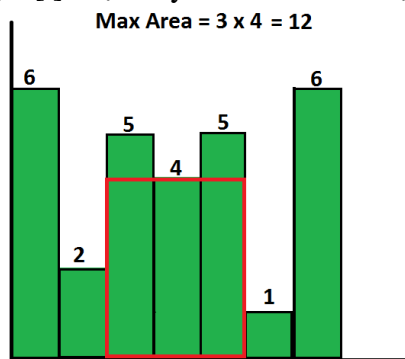
Với mỗi test, in ra trên một dòng N số $R[i]$, với $R[i]$ là giá trị phần tử đầu tiên lớn hơn $A[i]$.

Ví dụ

Input	Output
3	5 25 25 -1
4	-1 -1 -1
4 5 2 25	5 5 -1 -1
3	
2 2 2	
4	
4 4 5 5	

L. HÌNH CHỮ NHẬT LỚN NHẤT

Cho N cột, mỗi cột có chiều cao bằng $H[i]$. Bạn hãy tìm hình chữ nhật lớn nhất bị che phủ bởi các cột?



Input:

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

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

Dòng tiếp theo gồm N số nguyên $H[i]$ ($1 \leq H[i] \leq 10^9$).

Output:

Với mỗi test, in ra diện tích hình chữ nhật lớn nhất tìm được.

Ví dụ:

Input	Output
2	12
7	6
6 2 5 4 5 1 6	
3	
2 2 2	

M. HÌNH CHỮ NHẬT 0-1

Cho một bảng kích thước $N \times M$, được chia thành lưới ô vuông đơn vị N dòng M cột. Trên các ô của bảng ghi số 0 hoặc 1. Các dòng của bảng được đánh số 1, 2... N theo thứ tự từ trên xuống dưới và các cột của bảng được đánh số 1, 2..., M theo thứ tự từ trái qua phải

Yêu cầu: Hãy tìm một hình chữ nhật gồm các ô của bảng thoả mãn các điều kiện sau:

- 1 - Hình chữ nhật đó chỉ gồm các số 1
- 2 - Cạnh hình chữ nhật song song với cạnh bảng
- 3 - Diện tích hình chữ nhật là lớn nhất có thể

Input:

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

Mỗi test bắt đầu bởi hai số nguyên N và M ($1 \leq M, N \leq 500$).

N dòng tiếp theo, mỗi dòng gồm M số nguyên 0 hoặc 1.

Output:

Với mỗi test, in ra diện tích hình chữ nhật lớn nhất tìm được.

Ví dụ:

Input	Output
1 11 13 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 1	49

N. TRÒ CHƠI VÒNG TRÒN

(Đề xuất bởi Trần Thanh Sơn)

Có N người đứng thành một vòng tròn. Mỗi người được đánh thứ tự từ 1 đến N theo chiều kim đồng hồ. Trò chơi như sau: Ban đầu một số nguyên M được chọn. Mọi người bắt đầu đếm liên tiếp từ 1 đến M , bắt đầu từ người thứ nhất theo chiều kim đồng hồ. Sau khi số M được đếm thì người tiếp theo sẽ bị loại khỏi vòng, và người tiếp theo người bị loại sẽ tiếp tục đếm từ 1. Trò chơi dừng lại khi chỉ còn duy nhất một người

Cho hai số nguyên N và M , hãy tìm ra thứ tự của người chơi cuối cùng trong trò chơi:

Input

Dòng đầu tiên gồm 1 số nguyên T ($1 \leq T \leq 20$) là số lượng bộ test.

Tiếp theo là T dòng, mỗi dòng gồm 2 số nguyên N, M ($1 \leq N \leq 5000, 1 \leq M \leq 10000$).

Output

Với mỗi bộ test, in ra trên một dòng kết quả là thứ tự của người chơi cuối cùng.

Ví dụ

Input	Output
2	2
5 4	1
6 4	

O. NHỊP CHỨNG KHOÁN

(Đề xuất bởi *Trần Thanh Sơn*)

Bạn là một nhà đầu tư chứng khoán nổi tiếng. Nhiệm vụ hàng ngày của bạn là tính nhịp tăng giảm của phiên chứng khoán trong N ngày để có thể bắt kịp thị trường. Nhịp chứng khoán của ngày thứ i được định nghĩa là số ngày liên tiếp từ ngày thứ i trở về mà giá chứng khoán bé hơn hoặc bằng với giá chứng khoán của ngày i .

Input

Dòng đầu tiên gồm 1 số nguyên N ($1 \leq N \leq 10^5$) là số ngày.

Dòng tiếp theo gồm N số nguyên A_1, A_2, \dots, A_N ($1 \leq A_i \leq 10^6$) là giá chứng khoán của các ngày.

Output

In ra N số B_1, B_2, \dots, B_N trong đó B_i là nhịp chứng khoán của ngày thứ i .

Ví dụ:

Input	Output
7	1 1 1 2 1 4 6
100 80 60 70 60 75 85	