

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

CONTEST 7 – NGĂN XẾP

BÀI 1. 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	

BÀI 2. 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	

BÀI 3. 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+*

BÀI 4. 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 chuỗi 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-

BÀI 5. DÃY NGOẶC ĐÚNG DÀI NHẤT

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

- Chuỗi 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 chuỗi S. Nhiệm vụ của bạn là hãy tìm dãy ngoặc đúng dài nhất xuất hiện trong chuỗi đã cho.

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

Mỗi test gồm một chuỗi 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

BÀI 6. KIỂM TRA DẪY NGOẶC ĐÚNG

Cho một chuỗi chỉ gồm các ký tự '(', ')', '[', ']', '{', '}'. Một dãy ngoặc đúng được định nghĩa như sau:

- Chuỗi 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 chuỗi S. Nhiệm vụ của bạn là xác định chuỗi 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 chuỗi 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
[()]	

BÀI 7. SỬA LẠI DẪY NGOẶC

Cho một chuỗi chỉ gồm các ký tự '(', ')', '[', ']' và có độ dài chẵn. Hãy đếm số lượng dấu ngoặc cần phải đổi chiều ít nhất, sao cho chuỗi 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 chuỗi 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
) (()) (((

BÀI 8. 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 ký tự, có chứa ít nhất 1 và không quá 10 cặp dấu ngoặc.

Kết quả: In ra tất các 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

BÀI 9. 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	16
6*3+2- (6-4/2)	2278
100+99*22	102
6* ((4*3) +5)	-1
1-2	

BÀI 10. 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].

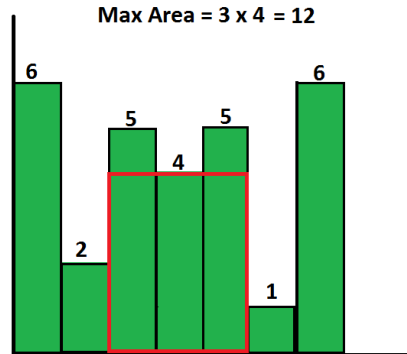
Ví dụ

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	

BÀI 11. 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	

BÀI 12. BIẾN ĐỔI TIỀN TỔ - TRUNG TỔ

Có ba dạng biểu diễn cho các biểu thức số học và logic:

Infix (trung tố): Biểu diễn biểu thức dưới dạng trung tố là phép biểu diễn biểu thức trong đó phép toán được đặt giữa hai toán hạng. Ví dụ $(A+B) * (C-D)$.

Prefix (tiền tố): Biểu diễn biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ $*+AB-CD$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Postfix (hậu tố): Biểu diễn biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ $AB+CD-*$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng tiền tố về dạng trung tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 10^6$.

Input	Output
2	$((A+B) * (C-D))$
$*+AB-CD$	$((A - (B/C)) * ((A/K) - L))$
$*-A/BC-/AKL$	

BÀI 13. BIẾN ĐỔI TIỀN TỐ - HẬU TỐ

Có ba dạng biểu diễn cho các biểu thức số học và logic:

Infix (trung tố): Biểu diễn biểu thức dưới dạng trung tố là phép biểu diễn biểu thức trong đó phép toán được đặt giữa hai toán hạng. Ví dụ $(A+B) * (C-D)$.

Prefix (tiền tố): Biểu diễn biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ $*+AB-CD$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Postfix (hậu tố): Biểu diễn biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ $AB+CD-*$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng tiền tố về dạng hậu tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 10^6$.

Input	Output
2	$AB+CD-*$
$*+AB-CD$	$ABC/-AK/L-*$

BÀI 14. BIẾN ĐỔI HẬU TỔ - TIỀN TỔ

Có ba dạng biểu diễn cho các biểu thức số học và logic:

Infix (trung tố): Biểu diễn biểu thức dưới dạng trung tố là phép biểu diễn biểu thức trong đó phép toán được đặt giữa hai toán hạng. Ví dụ $(A+B) * (C-D)$.

Prefix (tiền tố): Biểu diễn biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ $*+AB-CD$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Postfix (hậu tố): Biểu diễn biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ $AB+CD-*$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng hậu tố về dạng tiền tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 10^6$.

Input	Output
2	*+AB-CD
AB+CD-*	*-A/BC-/AKL
ABC/-AK/L-*	

BÀI 15. BIẾN ĐỔI HẬU TỔ - TRUNG TỔ

Có ba dạng biểu diễn cho các biểu thức số học và logic:

Infix (trung tố): Biểu diễn biểu thức dưới dạng trung tố là phép biểu diễn biểu thức trong đó phép toán được đặt giữa hai toán hạng. Ví dụ $(A+B) * (C-D)$.

Prefix (tiền tố): Biểu diễn biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ $*+AB-CD$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Postfix (hậu tố): Biểu diễn biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ $AB+CD-*$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng hậu tố về dạng trung tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ;

- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.
- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 10^6$.

Input	Output
2	(A+ (B+C)
ABC++	((A*B) +C)
AB*C+	

BÀI 16. BIẾN ĐỔI TRUNG TỐ - HẬU TỐ

Có ba dạng biểu diễn cho các biểu thức số học và logic:

Infix (trung tố): Biểu diễn biểu thức dưới dạng trung tố là phép biểu diễn biểu thức trong đó phép toán được đặt giữa hai toán hạng. Ví dụ (A+B) * (C-D).

Prefix (tiền tố): Biểu diễn biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ *+AB-CD (tương ứng với biểu thức trung tố (A+B)*(C-D)).

Postfix (hậu tố): Biểu diễn biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ AB+CD-* (tương ứng với biểu thức trung tố (A+B)*(C-D)).

Hãy viết chương trình chuyển đổi biểu thức biểu diễn dưới dạng trung tố về dạng hậu tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 10$.

Input	Output
2	ABC++
(A+ (B+C)	AB*C+
((A*B) +C)	

BÀI 17. TÍNH GIÁ TRỊ BIỂU THỨC TIỀN TỐ

Có ba dạng biểu diễn cho các biểu thức số học và logic:

Infix (trung tố): Biểu diễn biểu thức dưới dạng trung tố là phép biểu diễn biểu thức trong đó phép toán được đặt giữa hai toán hạng. Ví dụ (A+B) * (C-D).

Prefix (tiền tố): Biểu diễn biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ $*+AB-CD$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Postfix (hậu tố): Biểu diễn biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ $AB+CD-*$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Hãy viết chương trình chuyển tính toán giá trị của biểu thức tiền tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp. Các số xuất hiện trong biểu thức là các số đơn có 1 chữ số.

Output:

- Đưa ra kết quả mỗi test theo từng dòng, chỉ lấy giá trị phần nguyên.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 20$.

Input	Output
2	8
$-+8/632$	25
$-+7*45+20$	

BÀI 18. TÍNH GIÁ TRỊ BIỂU THỨC HẬU TỐ

Có ba dạng biểu diễn cho các biểu thức số học và logic:

Infix (trung tố): Biểu diễn biểu thức dưới dạng trung tố là phép biểu diễn biểu thức trong đó phép toán được đặt giữa hai toán hạng. Ví dụ $(A+B) * (C-D)$.

Prefix (tiền tố): Biểu diễn biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ $*+AB-CD$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Postfix (hậu tố): Biểu diễn biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ $AB+CD-*$ (tương ứng với biểu thức trung tố $(A+B)*(C-D)$).

Hãy viết chương trình chuyển tính toán giá trị của biểu thức hậu tố.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức hậu tố exp. Các số xuất hiện trong biểu thức là các số đơn có 1 chữ số.

Output:

- Đưa ra kết quả mỗi test theo từng dòng, chỉ lấy giá trị phần nguyên.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{lenght}(\text{exp}) \leq 20$.

Input	Output
2	-4

231*+9- 875*+9-	34
--------------------	----

BÀI 19. KIỂM TRA BIỂU THỨC SỐ HỌC

Cho biểu thức số học, hãy cho biết biểu thức số học có dư thừa các cặp ký hiệu ‘(,’) ‘ hay không?

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức tiền tố exp.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, exp thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{length}(\text{exp}) \leq 20$.

Input	Output
3	Yes
((a+b))	Yes
(a + (b) / c)	No
(a + b* (c-d))	

BÀI 20. ĐẢO TỪ

Cho một chuỗi ký tự str bao gồm nhiều từ trong chuỗi. Hãy đảo ngược từng từ trong chuỗi?

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một dòng ghi lại nhiều từ trong chuỗi str.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, str thỏa mãn ràng buộc: $1 \leq T \leq 100$; $2 \leq \text{length}(\text{str}) \leq 10^6$.

Input	Output
2	CBA FED
ABC DEF	321 654
123 456	

BÀI 21. SO SÁNH BIỂU THỨC

Cho P1, P2 là hai biểu thức đúng chỉ bao gồm các ký tự mở ngoặc ‘(’ hoặc đóng ngoặc ‘)’ và các toán hạng in thường. Nhiệm vụ của bạn là định xem P1 và P2 có giống nhau hay không.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào P1, dòng tiếp theo đưa vào P2.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, P thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{length}(P) \leq 100$.

Input	Output
2	YES
$-(a+b+c)$	NO
$-a-b-c$	
$a-b-(c-d)$	
$a-b-c-d$	

BÀI 22. BIỂU THỨC TĂNG GIẢM

Cho mảng A[] chỉ bao gồm các ký tự I hoặc D. Ký tự I được hiểu là tăng (Increasing) ký tự D được hiểu là giảm (Decreasing). Sử dụng các số từ 1 đến 9, hãy đưa ra số nhỏ nhất được đoán nhận từ mảng A[]. Chú ý, các số không được phép lặp lại. Dưới đây là một số ví dụ mẫu:

- A[] = "I" : số tăng nhỏ nhất là 12.
- A[] = "D" : số giảm nhỏ nhất là 21
- A[] = "DD" : số giảm nhỏ nhất là 321
- A[] = "DDIDDIID": số thỏa mãn 321654798

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ID
- T, Length(A) thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(A) \leq 8$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
4	12
I	21
D	321
DD	321654798
DDIDDIID	

BÀI 23. PHẦN TỬ BÊN PHẢI NHỎ HƠN

Cho mảng A[] gồm n phần tử. Hãy đưa ra các phần tử nhỏ hơn tiếp theo của phần tử lớn hơn đầu tiên phần tử hiện tại. Nếu phần tử hiện tại không có phần tử lớn hơn tiếp theo ta xem là -1. Nếu phần tử không có phần tử nhỏ hơn tiếp theo ta cũng xem là -1. Ví dụ với mảng A[] = {5, 1, 9, 2, 5, 1, 7} ta có kết quả là ans = {2, 2, -1, 1, -1, -1, -1} vì:

Next Greater	Right Smaller
5 -> 9	9 -> 2
1 -> 9	9 -> 2
9 -> -1	-1 -> -1
2 -> 5	5 -> 1

5	-> 7	7	-> -1
1	-> 7	7	-> -1
7	-> -1	7	-> -1

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test gồm hai dòng: dòng thứ nhất đưa vào n là số phần tử của mảng A[], dòng tiếp theo đưa vào n số A[i].

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n$, $A[i] \leq 10^6$.

Input	Output
2	2 2 1 1 -1 -1 -1
7	2 5 5 5 -1 3 -1 -1
5 1 9 2 5 1 7	
8	
4 8 2 1 9 5 6 3	

BÀI 24. GIẢI MÃ XÂU KÝ TỰ

Cho chuỗi ký tự mã hóa str. Hãy viết chương trình giải mã chuỗi ký tự str. Chuỗi ký tự mã hóa được thực hiện theo số lần lặp các chuỗi con của str như sau:

Xâu đầu vào: "abbbababbbababbbab "

Xâu mã hóa : "3[a3[b]1[ab]]"

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một chuỗi mã hóa str được viết trên một dòng.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, str thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{length}(\text{str}) \leq 100$.

Input	Output
2	b
1 [b]	bcacabcbacabcbaca
3 [b2 [ca]]	

BÀI 25. BỎ DẤU NGOẶC

Cho biểu thức đúng P chỉ bao gồm các phép toán +, -, các toán hạng cùng với các ký tự '(', ')'. Hãy bỏ tất cả các ký tự '(', ')' trong P để nhận được biểu thức tương đương. Ví dụ với $P = a - (b + c)$ ta có kết quả $P = a - b - c$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T;
- Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức P được viết trên một dòng.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Ràng buộc:

- T, P thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{length}(P) \leq 10^3$.

Input	Output
2	a-b-c
a- (b+c)	a-b+c+d+e-f
a- (b-c- (d+e)) -f	