Dãy Ngoặc đúng dài nhất

Cho một xâu chỉ gồm các kí 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 ≤ 20).

Mỗi test gồm một xâu S có độ dài không vượt quá 105 kí 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 |

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

main(){

int t;

cin >> t;

while(t--){

string s;

cin >> s;

stack<int> st;

st.push(-1);

int res = 0;

for(int i = 0; i < s.size(); i++){

if(s[i] == '(') st.push(i);

else{

st.pop();

if(st.empty()) st.push(i);

else{

res = max(res, i-st.top());

}

}

}

cout << res << endl;

}

}

NGĂN XẾP 1

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07001/top)

Cho một ngăn xếp các số nguyên. Các thao tác 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.

**Input:** 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.

**Output:** 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  push 5  show  push 7  show  pop  pop  show | 3 5  3 5 7  3 |

#include <bits/stdc++.h>

using namespace std;

int a[205], top = -1;

void push(int n) {

top++;

a[top] = n;

}

void pop() {

top--;

}

void show() {

if(top >= 0) {

for(int i=0; i<=top; i++){

cout << a[i] <<" ";

}

cout << endl;

}

else {

cout << "empty" << endl;

}

}

int main() {

string s;

int n;

while(cin >> s){

if(s == "push"){

cin >> n;

push(n);

}

else if(s == "pop") {

pop();

}

else if(s == "show") {

show();

}

}

}

NGĂN XẾP 2

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07002/top)

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

“PUSH x”: Thêm phần tử x vào stack (0 ≤ x ≤ 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 ≤ 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  PUSH 1  PUSH 2  POP  PRINT  PUSH 3  PRINT  POP  POP  PRINT | 1  3  NONE |

#include<bits/stdc++.h>

using namespace std;

int main(){

int T; cin >> T;

stack<int> S;

string str; int n;

while (T--){

cin >> str;

if (str == "PUSH"){

cin >> n;

S.push(n);

}

else if (str == "PRINT"){

if (S.empty()) cout << "NONE" << endl;

else cout << S.top() << endl;

}

else if (str == "POP"){

if (!S.empty()) S.pop();

}

}

}

 KIỂM TRA BIỂU THỨC SỐ HỌC

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07003/top)

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≤T≤100; 2≤length(exp)≤20.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

bool solve(string &str){

stack<char> s;

for (int i=0; i<str.length(); i++){

if (str[i] == ')'){

char top = s.top();

s.pop();

bool flag = true;

while (top != '('){

if (top=='+' || top=='-' || top=='\*' || top=='/')

flag = false;

top = s.top();

s.pop();

}

if (flag == true)

return true;

}

else

s.push(str[i]);

}

return false;

}

int main(){

int T; cin >> T;

while (T--){

string s; cin >> s;

bool ans = solve(s);

if (ans == true) cout << "Yes" << endl;

else cout << "No" << endl;

}

}

 ĐẾM SỐ DẤU NGOẶC ĐỔI CHIỀU

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07004/top)

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 ≤ 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 |

#include<bits/stdc++.h>

using namespace std;

void solve(string str){

stack<char> s;

for (int i=0; i<str.length(); i++){

if (str[i]==')' && !s.empty()){

if (s.top() == '(')

s.pop();

else

s.push(str[i]);

}

else

s.push(str[i]);

}

int m = s.size();

int n = 0;

while (!s.empty() && s.top() == '('){

s.pop();

n++;

}

int res = m/2+n%2;

cout << res << endl;

}

int main(){

int T; cin >> T;

while (T--){

string str; cin >> str;

solve(str);

}

}

BIẾN ĐỔI TIỀN TỐ - TRUNG TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07009/top)

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≤T≤100; 2≤length(exp)≤106.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

bool isOperator(char c){

if (c == '\*' || c == '/' || c == '+' || c == '-')

return true;

return false;

}

void solve(string str){

stack<string> s;

for (int i = str.length()-1; i >= 0; i--){

if (isOperator(str[i])){

string str1 = s.top(); s.pop();

string str2 = s.top(); s.pop();

string tmp = "(" + str1 + str[i] + str2 + ")";

s.push(tmp);

}

else

s.push(string(1, str[i]));

}

cout << s.top() << endl;

}

int main() {

int t; cin >> t;

while (t--){

string s; cin >> s;

solve(s);

}

}

 BIẾN ĐỐI TIỀN TỐ - HẬU TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07010/top)

   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≤T≤100; 2≤length(exp)≤106.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

bool isOperator(char c){

if (c == '\*' || c == '/' || c == '+' || c == '-')

return true;

return false;

}

void solve(string str) {

stack<string> s;

int length = str.size();

for (int i = length - 1; i >= 0; i--) {

if (isOperator(str[i])) {

string op1 = s.top(); s.pop();

string op2 = s.top(); s.pop();

string temp = op1 + op2 + str[i];

s.push(temp);

}

else {

s.push(string(1, str[i]));

}

}

cout << s.top() << endl;

}

int main() {

int t; cin >> t;

cin.ignore();

while (t--){

string s; getline(cin,s);

solve(s);

}

}

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≤T≤100; 2≤length(exp)≤20.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  231\*+9–  875\*+9- | -4  34 |

#include<bits/stdc++.h>

using namespace std;

bool isOperator(char c){

if (c == '\*' || c == '/' || c == '+' || c == '-')

return true;

return false;

}

void solve(string str){

stack<int> s;

for (int i = 0; i < str.length(); i++){

if (isOperator(str[i])){

int x1 = s.top(); s.pop();

int x2 = s.top(); s.pop();

switch (str[i]){

case '+': s.push(x2 + x1); break;

case '-': s.push(x2 - x1); break;

case '\*': s.push(x2 \* x1); break;

case '/': s.push(x2 / x1); break;

}

}

else{

s.push(str[i] - '0');

}

}

cout << s.top() << endl;

}

int main() {

int t; cin >> t;

cin.ignore();

while (t--){

string s; getline(cin,s);

solve(s);

}

}

 TÍNH GIÁ TRỊ BIỂU THỨC TIỀN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07014/top)

Hãy viết chương trình 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≤T≤100; 2≤length(exp)≤20.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

bool isOperator(char c){

if (c == '\*' || c == '/' || c == '+' || c == '-')

return true;

return false;

}

void solve(string str){

stack<int> s;

for (int i = str.length()-1; i >=0; i--){

if (isOperator(str[i])){

int x1 = s.top(); s.pop();

int x2 = s.top(); s.pop();

switch (str[i]){

case '+': s.push(x1 + x2); break;

case '-': s.push(x1 - x2); break;

case '\*': s.push(x1 \* x2); break;

case '/': s.push(x1 / x2); break;

}

}

else{

s.push(str[i] - '0');

}

}

cout << s.top() << endl;

}

int main(){

int t; cin >> t;

cin.ignore();

while (t--){

string s; getline(cin,s);

solve(s);

}

}

PHẦN TỬ BÊN PHẢI ĐẦU TIÊN LỚN HƠN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07027/top)

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 ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (1 ≤ N ≤ 100000).

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 109).

**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  4  4 5 2 25  3  2 2 2  4  4 4 5 5 | 5 25 25 -1  -1 -1 -1  5 5 -1 -1 |

#include<bits/stdc++.h>

typedef long long ll;

using namespace std;

void solve(ll arr[], int n){

stack<ll> s;

ll ans[n];

for (int i = n-1; i >= 0; i--){

while (!s.empty() && arr[i] >= s.top())

s.pop();

if (s.empty())

ans[i] = -1;

else

ans[i] = s.top();

s.push(arr[i]);

}

for (int i = 0; i < n; i++)

cout << ans[i] << " ";

cout << endl;

}

int main(){

int t; cin >> t ;

while (t--){

int n; cin >> n;

ll \*arr = new ll[n];

for (int i = 0; i < n; i++) cin >> arr[i];

solve(arr, n);

delete arr;

}

}

 HÌNH CHỮ NHẬT LỚN NHẤT

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07019/top)

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?

Chart, histogram

Description automatically generated

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (N ≤ 100 000).

Dòng tiếp theo gồm N số nguyên H[i] (1 ≤ H[i] ≤ 109).

**Output:**

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

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  7  6 2 5 4 5 1 6  3  2 2 2 | 12  6 |

#include<bits/stdc++.h>

typedef long long ll;

using namespace std;

void solve(ll arr[], int n){

stack<int> s;

ll res = 0, tp, area;

int i = 0;

while (i < n){

if (s.empty() || arr[s.top()] < arr[i])

s.push(i++);

else{

tp = s.top(); s.pop();

area = arr[tp] \* (s.empty()? i: i-s.top()-1);

if (area > res)

res = area;

}

}

while (!s.empty()){

tp = s.top(); s.pop();

area = arr[tp] \* (s.empty()? i: i-s.top()-1);

if (area > res)

res = area;

}

cout << res << endl;

}

int main(){

int t; cin >> t ;

while (t--){

int n; cin >> n;

ll \*arr = new ll[n];

for (int i = 0; i < n; i++) cin >> arr[i];

solve(arr, n);

delete arr;

}

}

BIẾN ĐỔI TRUNG TỐ - HẬU TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07008/top)

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≤T≤100; 2≤length(exp)≤10.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

int prec(char c){

if(c == '^')

return 3;

else if(c == '\*' || c == '/')

return 2;

else if(c == '+' || c == '-')

return 1;

else

return -1;

}

void solve(string s){

stack<char> st;

int l = s.length();

string ns;

for(int i = 0; i < l; i++)

{

if((s[i] >= 'a' && s[i] <= 'z')||(s[i] >= 'A' && s[i] <= 'Z'))

ns+=s[i];

else if(s[i] == '(')

st.push('(');

else if(s[i] == ')')

{

while(!st.empty() && st.top() != '(')

{

char c = st.top();

st.pop();

ns += c;

}

if(st.top() == '(')

st.pop();

}

else{

while(!st.empty() && prec(s[i]) <= prec(st.top()))

{

char c = st.top();

st.pop();

ns += c;

}

st.push(s[i]);

}

}

while(!st.empty())

{

char c = st.top();

st.pop();

if (c != '(')

ns += c;

}

cout << ns << endl;

}

int main() {

int t; cin >> t;

cin.ignore();

while (t--){

string s; cin >> s;

solve(s);

}

return 0;

}

BIẾN ĐỔI HẬU TỐ - TIỀN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07011/top)

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≤T≤100; 2≤length(exp)≤106.

**Ví  dụ:**

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

#include<bits/stdc++.h>

using namespace std;

bool isOperator(char c){

if (c == '\*' || c == '/' || c == '+' || c == '-')

return true;

return false;

}

void solve(string str){

stack<string> s;

for (int i = 0; i < str.length(); i++){

if (isOperator(str[i])){

string str1 = s.top(); s.pop();

string str2 = s.top(); s.pop();

string tmp = str[i] + str2 + str1;

s.push(tmp);

}

else

s.push(string(1, str[i]));

}

cout << s.top() << endl;

}

int main() {

int t; cin >> t;

cin.ignore();

while (t--){

string s; getline(cin,s);

solve(s);

}

}

 BIẾN ĐỔI HẬU TỐ - TIỀN TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07011/top)

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≤T≤100; 2≤length(exp)≤106.

**Ví  dụ:**

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

#include<bits/stdc++.h>

using namespace std;

bool isOperator(char c){

if (c == '\*' || c == '/' || c == '+' || c == '-')

return true;

return false;

}

void solve(string str){

stack<string> s;

for (int i = 0; i < str.length(); i++){

if (isOperator(str[i])){

string str1 = s.top(); s.pop();

string str2 = s.top(); s.pop();

string tmp = str[i] + str2 + str1;

s.push(tmp);

}

else

s.push(string(1, str[i]));

}

cout << s.top() << endl;

}

int main() {

int t; cin >> t;

cin.ignore();

while (t--){

string s; getline(cin,s);

solve(s);

}

}

BIẾN ĐỔI HẬU TỐ - TRUNG TỐ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07012/top)

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 màng ràng buộc: 1≤T≤100; 2≤length(exp)≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc**:

* T, exp thỏa mãn ràng buộc: 1≤T≤100; 2≤length(exp)≤106.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

bool isOperator(char c){

if (c == '\*' || c == '/' || c == '+' || c == '-')

return true;

return false;

}

void solve(string str){

stack<string> s;

for (int i = 0; i < str.length(); i++){

if (isOperator(str[i])){

string str1 = s.top(); s.pop();

string str2 = s.top(); s.pop();

string tmp = "(" + str2 + str[i] + str1 + ")";

s.push(tmp);

}

else

s.push(string(1, str[i]));

}

cout << s.top() << endl;

}

int main() {

int t; cin >> t;

cin.ignore();

while (t--){

string s; getline(cin,s);

solve(s);

}

}

BIỂU THỨC TĂNG GIẢM

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07016/top)

Cho dãy ký tự S 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 S. 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 S
* T, S thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ length(S) ≤8; .

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

main(){

int t;

cin >> t;

while(t--){

string s;

cin >> s;

stack<int> st;

string res = "";

for(int i = 0; i <= s.size(); i++){

st.push(i+1);

if(s[i] == 'I' || i == s.size()){

while(!st.empty()){

res += (st.top()+'0');

st.pop();

}

}

}

cout << res << endl;

}

}

PHẦN TỬ BÊN PHẢI NHỎ HƠN

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07017/top)

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≤T≤100; 1≤n, A[i] ≤106.

**Ví dụ:**

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

#include<bits/stdc++.h>

using namespace std;

void nextGreater(int arr[], int n, int next[], char order){

stack<int> S;

for (int i=n-1; i>=0; i--){

while (!S.empty() && ((order=='G')? arr[S.top()] <= arr[i]: arr[S.top()] >= arr[i]))

S.pop();

if (S.empty())

next[i] = -1;

else

next[i] = S.top();

S.push(i);

}

}

void solve(int arr[], int n)

{

int NG[n];

int RS[n];

nextGreater(arr, n, NG, 'G');

nextGreater(arr, n, RS, 'S');

for (int i=0; i< n; i++){

if (NG[i] != -1 && RS[NG[i]] != -1)

cout << arr[RS[NG[i]]] << " ";

else

cout<<"-1"<<" ";

}

cout << endl;

}

int main(){

int t; cin >> t ;

while (t--){

int n; cin >> n;

int \*arr = new int[n];

for (int i = 0; i < n; i++) cin >> arr[i];

solve(arr, n);

delete arr;

}

}

GIẢI MÃ XÂU KÝ TỰ

[Bài làm tốt nhất](https://code.ptit.edu.vn/student/question/DSA07029/top)

Cho xâu ký tự mã hóa str. Hãy viết chương trình giải mã xâu ký tự str. Xâu ký tự mã hóa được thực hiện theo số lần lặp các xâu 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 xâu 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≤T≤100; 1≤length(str)≤100.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 1[b] 3[b2[ca]] | b bcacabcacabcaca |

#include<bits/stdc++.h>

using namespace std;

void solve(string str){

stack<int> ig;

stack<char> sg;

string tmp = "", res = "";

for (int i = 0; i < str.length(); i++){

int num = 0;

if (str[i] >= '0' && str[i] <= '9'){

while (str[i] >= '0' && str[i] <= '9'){

num = num\*10 + str[i] - '0';

i++;

}

i--;

ig.push(num);

}

else if (str[i] == ']'){

tmp = "";

num = 0;

if (!ig.empty()){

num = ig.top();

ig.pop();

}

while (!sg.empty() && sg.top() != '['){

tmp = sg.top() + tmp;

sg.pop();

}

if (!sg.empty() && sg.top() == '[')

sg.pop();

for (int j = 0; j < num; j++)

res = res + tmp;

for (int j = 0; j < res.length(); j++)

sg.push(res[j]);

res = "";

}

else if (str[i] == '['){

if (str[i-1] >= '0' && str[i-1] <= '9')

sg.push(str[i]);

else{

sg.push(str[i]);

ig.push(1);

}

}

else

sg.push(str[i]);

}

while (!sg.empty()){

res = sg.top() + res;

sg.pop();

}

cout << res << endl;

}

int main(){

int t; cin >> t;

while (t--){

string str; cin >> str;

solve(str);

}

}