Pilha
Uma pilha é uma das várias estruturas de dados que admitem remoção de elementos e inserção de novos elementos. Mais especificamente, uma *pilha (= stack)* é uma estrutura sujeita à seguinte regra de operação: sempre que houver uma remoção, o elemento removido é o que está na estrutura há menos tempo.

Em outras palavras, o primeiro objeto a ser inserido na pilha é o último a ser removido. Essa política é conhecida pela sigla LIFO *(= Last-In-First-Out)*.

APLICAÇÃO BUSCA EM PROFUNDIDADE EM C

```
#include <stdio.h>
#define MAX 101
int pilha[MAX], d[MAX], g[MAX][MAX], marc[MAX],topo;
int inicializa(){
	topo = 0;
}
void push(int x){
	pilha[topo++] = x;
}
int pop(){
	return pilha[--topo];
}
int vazia(){
	return topo==0;
}
int main(){
	int n,m,u,i,v,a,b,cont,teste=1;
	while(1){
		scanf("%d %d",&n,&m);
		if(n==0 && m==0) break;
		for(i=1;i<=n;i++){
		 d[i] = 0;
		 marc[i]=0;
		}
		for(i=1;i<=m;i++){
			scanf("%d %d",&a,&b);
			g[a][d[a]++]=b;
			g[b][d[b]++]=a;
		}
		inicializa();
		marc[1]=1;
		push(1);
		cont = 0;
		while(!vazia()){
			u = pop();
			cont++;

			for(i=0;i<d[u];i++){
				v = g[u][i];
				if(marc[v]==0){
					marc[v]=1;
					push(v);
				}
			}
		}
		printf("Teste %d\n",teste++);
		if(cont==n) printf("normal\n\n");
		else printf("falha\n\n");

	}

	return 0;


}
```

OBS: 2009-10-12 14:53:59 Transmissão de Energia aceito 0.02 1.7M C

IMPLEMENTAÇÃO C++

```
#include <iostream>
#include <vector>
#include <stack>
#define MAX 101
using namespace std;
vector <int> lista[MAX];
vector <int>::iterator it;
stack <int> pilha;
bool marc[MAX];
int main(){
	int n,m,u,v,i,a,b,cont,teste=1;
	while(true){
		cin >> n >> m;
		if(n==0 && m==0) break;
		for(i=1;i<=n;i++){
			 marc[i]=false;
			lista[i].clear();
		}
		for(i=1;i<=m;i++){
			cin >> a >> b;
			lista[a].push_back(b);
			lista[b].push_back(a);
		}
		marc[1] = true;
		pilha.push(1);
		cont = 0;
		while(!pilha.empty()){
			u = pilha.top();
			pilha.pop();
			cont++;
			for(it = lista[u].begin(); it!=lista[u].end(); it++){
				if(!marc[*it]){
					marc[*it]=true;
					pilha.push(*it);
				}
			}
		}
		cout << "Teste " << teste++ << endl;
		if(cont==n)
		 cout << "normal" << endl;
		else
		 cout << "falha" << endl;
		cout << endl;
	}
}
```

OBS: 2009-10-12 15:26:37 Transmissão de Energia aceito 0.13 2.6M C++

Exercício

https://br.spoj.pl/problems/ORKUT/
https://br.spoj.pl/problems/PEDAGIO/
https://br.spoj.pl/problems/CIRCUITO/
https://br.spoj.pl/problems/NUMERDOS/
https://br.spoj.pl/problems/ODDOREVE/
https://br.spoj.pl/problems/PREEMPOS/
https://br.spoj.pl/problems/CONTAGEM/

673- Parentheses Balance

You are given a string consisting of parentheses `()` and `[]`. A string of this type is said to be *correct*:

(a) if it is the empty string

(b) if A and B are correct, AB is correct,

(c) if A is correct, `(A)` and `[A]` is correct.

Write a program that takes a sequence of strings of this type and check their correctness. Your program can assume that the maximum string length is 128.

Input

The file contains a positive integer *n* and a sequence of *n* strings of parentheses `()` and `[]`, one string a line.

Output

A sequence of `Yes` or `No` on the output file.

Sample Input

```
3
([])
((([()])))
([()[]()])()
```

Sample Output

```
Yes
No
Yes
```