#include<bits/stdc++.h>

using namespace std;

void topo(char src, unordered\_map<char,vector<char> > graph, stack<char> &st, vector<bool> &vis){

vis[src-'a'] = true;

auto it = graph[src];

for(auto ch: it){

if(vis[ch-'a']==false){

topo(ch,graph,st,vis);

}

}

st.push(src);

}

string solve(vector<string> &v) {

unordered\_map<char,vector<char> > graph;

unordered\_map<char,int> mp;

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

string s = v[i];

for(auto it: s){

mp[it]++;

}

}

int n = v.size();

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

string s1 = v[i];

string s2 = v[i+1];

int len = s1.size()>s2.size()? s2.size(): s1.size();

for(int j=0;j<len;j++){

if(s1[j]!=s2[j]){

graph[s1[j]].push\_back(s2[j]);

break;

}

}

}

// now just make toposort of the graph

stack<char> st;

vector<bool> vis(26,false);

if(graph.size()>0){

for(auto it: graph){

if(vis[it.first-'a']==false){

topo(it.first,graph,st,vis);

}

}

}

string ans="";

while(st.size()>0){

ans+= st.top();

st.pop();

}

ans = ans.size()==mp.size()?ans: "";

return ans;

}

int main(){

int n;

cin>>n;

vector<string> v(n);

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

cin>>v[i];

}

cout<<solve(v);

}