#include<bits/stdc++.h>

using namespace std;

void solve(vector<int> &arr, int n){

vector<int> dp(n);

dp[0] = arr[0];

int large = INT\_MIN,ind=0;

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

int max\_ = INT\_MIN;

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

if(arr[i]>arr[j]){

max\_ = max(max\_,dp[j]);

}

}

if(max\_==INT\_MIN){

dp[i] = 1;

}

else{

dp[i] = 1 + max\_;

}

// large = max(large,dp[i]);

if(large<dp[i]){

large = dp[i];

ind = i;

}

}

cout<<large<<endl;

int count = 0;

queue<pair<vector<int>,string> > q;

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

if(large==dp[i]){

string dd = to\_string(arr[i]);

q.push({{dp[i],i,arr[i]},dd+""});

}

}

while(!q.empty()){

pair<vector<int>, string> node = q.front();

q.pop();

vector<int> v = node.first; // v = {len,ind,val}

string s = node.second;

if(v[0]==1){

cout<<s<<endl;

}

for(int j=v[1]-1;j>=0;j--){

if(dp[j]==v[0]-1 && arr[j]<=v[2]){

string ss = to\_string(arr[j]);

q.push({{dp[j],j,arr[j]},ss+" -> "+s});

}

}

}

}

int main(){

int n;

cin>>n;

vector<int> v(n);

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

cin>>v[i];

}

solve(v,n);

}