**Reverse a Stack :-**

Medium Accuracy: 80.5% Submissions: 39K+ Points: 4

You are given a stack **St**. You have to reverse the stack using recursion.

**Example 1:**

**Input:**

St = {3,2,1,7,6}

**Output:**

{6,7,1,2,3}  
**Explanation:**  
Input stack after reversing will look like the stack in the output.

**Example 2:**

**Input:**

St = {4,3,9,6}

**Output:**

{6,9,3,4}  
**Explanation:**Input stack after reversing will look like the stack in the output.

**Your Task:**

You don't need to read input or print anything. Your task is to complete the function **Reverse()** which takes the stack **St**as input and reverses the given stack.

**Expected Time Complexity:** O(N)  
**Expected Auxiliary Space:** O(N)

**Constraints:**  
1 <= size of the stack <= 104  
-109 <= Each element of the stack <= 109  
Sum of N over all test cases doesn't exceeds 106  
Array may contain duplicate elements.

**Code :-**

//{ Driver Code Starts

//Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User function Template for C++

class Solution{

public:

queue<int> q;

void Reverse(stack<int> &St){

if(St.empty()) return;

q.push(St.top());

St.pop();

Reverse(St);

St.push(q.front());

q.pop();

return;

}

};

//{ Driver Code Starts.

int main(){

int T;

cin>>T;

while(T--){

int N;

cin>>N;

stack<int> St;

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

int x;

cin>>x;

St.push(x);

}

Solution ob;

ob.Reverse(St);

vector<int>res;

while(St.size())

{

res.push\_back(St.top());

St.pop();

}

for(int i = res.size()-1;i>=0;i--)

{

cout<<res[i]<<" ";

}

cout<<endl;

}

}

// } Driver Code Ends

**T.C :- O(N)**

**S.C :- O(N)**