**Reversing the equation :-**

Easy Accuracy: 58.61% Submissions: 20K+ Points: 2

Given a mathematical equation that contains only numbers and +, -, \*, /. Print the equation in reverse, such that the equation is reversed, but the numbers remain the same.  
It is guaranteed that the given equation is valid, and there are no leading zeros.

**Example 1:**

**Input:**

S = "20-3+5\*2"

**Output:** 2\*5+3-20

**Explanation**: The equation is reversed with

numbers remaining the same.

**Example 2:**

**Input**:

S = "5+2\*56-2/4"

**Output:** 4/2-56\*2+5

**Explanation**: The equation is reversed with

numbers remaining the same.

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **reverseEqn()**which takes the string S representing the equation as input and returns the resultant string representing the equation in reverse.

**Expected Time Complexity:**O(|S|).  
**Expected Auxiliary Space:**O(|S|).

**Constraints:**  
1<=|S|<=105  
The string contains only the characters '0' - '9', '+', '-', '\*', and '/'.

**Code :-**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution

{

public:

string reverseEqn (string &s){

int i=s.size()-1;

string ans;

while(i>=0){

//number is starting to fetch

if(s[i]!='+' && s[i]!='-' && s[i]!='/' && s[i]!='\*'){

int end=i, start;

while(s[i]!='+' && s[i]!='-' && s[i]!='/' && s[i]!='\*'){

start = i;

if(--i == -1)

break;

}

ans += s.substr(start, (end-start+1));

if(i<0)

return ans;

}

ans += s[i--];

}

return ans;

}

};

//{ Driver Code Starts.

int main()

{

int t; cin >> t;

while (t--)

{

string s; cin >> s;

Solution ob;

cout <<ob.reverseEqn (s) << endl;

}

}

// Contributed By: Pranay Bansal

// } Driver Code Ends

**T.C :- O( |S| )**

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