**Roman Number to Integer: -**

**Easy** Accuracy: **43.31%** Submissions: **143K+** Points: **2**

Given a string in roman no format (s)  your task is to convert it to an integer . Various symbols and their values are given below.  
I 1  
V 5  
X 10  
L 50  
C 100  
D 500  
M 1000

**Example 1:**

**Input:**

s = V

**Output:** 5

**Example 2:**

**Input:**

s = III

**Output:** 3

**Your Task:**  
Complete the function**romanToDecimal()** which takes a string as input parameter and returns the equivalent decimal number.

**Expected Time Complexity:**O(|S|), |S| = length of string S.  
**Expected Auxiliary Space:**O(1)

**Constraints:**  
1<=roman no range<=3999

**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:

unordered\_map<char,int> mp;

int romanToDecimal(string &str) {

// code here

// mapping character with their values

mp['I']=1; mp['V']=5; mp['X']=10;

mp['L']=50; mp['C']=100; mp['D']=500;

mp['M']=1000;

int n = str.size();

int i = 0, j, totalsum = 0, clustersum;

while(i < n){

// current cluster value

j = i;

clustersum = 0;

while(j < n and str[j] == str[i]){

clustersum += mp[str[j]];

++j;

}

// future letter has greater value

if(j < n and mp[str[j]] > mp[str[i]])

totalsum -= clustersum;

else

totalsum += clustersum;

i = j;

}

return totalsum;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin >> t;

while (t--) {

string s;

cin >> s;

Solution ob;

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

}

}

// } Driver Code Ends

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

**S.C: - O(1)**