**Palindrome String :-**

Easy Accuracy: 51.21% Submissions: 250K+ Points: 2

Given a string **S**, check if it is palindrome or not.

**Example 1:**

**Input:** S = "abba"

**Output:** 1

**Explanation:** S is a palindrome

**Example 2:**

**Input:** S = "abc"

**Output:** 0

**Explanation:** S is not a palindrome

**Your Task:**  
You don't need to read input or print anything. Complete the function **isPalindrome()**which accepts string S and returns an integer value 1 or 0.  
  
**Expected Time Complexity:**O(Length of S)  
**Expected Auxiliary Space:**O(1)

**Constraints:**  
1 <= Length of S<= 2\*105

**Code :-**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User function template for C++

class Solution{

public:

int isPalindrome(string &S){

int low=0, high=S.size()-1;

while(low <= high){

if(S[low] != S[high])

return 0;

++low;

--high;

}

return 1;

}

};

//{ Driver Code Starts.

int main()

{

ios\_base::sync\_with\_stdio(0);

cin.tie(NULL);

cout.tie(NULL);

int t;

cin >> t;

while(t--)

{

string s;

cin >> s;

Solution ob;

cout << ob.isPalindrome(s) << "\n";

}

return 0;

}

// } Driver Code Ends

**T.C :- ϴ(N)**

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