**Palindromic Ciphers**

Attempted by: **1121**

/

Accuracy: **92%**

/

Maximum Score: **20**

/

5 Votes

Tag(s):

Ad-Hoc, Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

Julius Cipher is a type of cipher which relates all the lowercase alphabets to their numerical position in the alphabet, i.e., value of a is 1, value of b is 2, value of z is 26 and similarly for the rest of them.

Little Chandan is obsessed with this Cipher and he keeps converting every single string he gets, to the final value one gets after the multiplication of the Julius Cipher values of a string. Arjit is fed up of Chandan's silly activities, so he decides to at least restrict Chandan's bad habits.

So, he asks Chandan not to touch any string which is a palindrome, otherwise he is allowed to find the product of the Julius Cipher values of the string.

**Input:**   
The first line contains *t* number of test cases. The first line of every test case contains a string, *S*.

**Output:**   
Print the **value** if the string is not a palindrome, otherwise print **Palindrome** - where **value** is equal to the product of all the characters of Julius Cipher values.

**Constraints:** 1 <= **T** <= 102  
1<=**length of the string**<=10

**Note:**  
The string will contain only lowercase letters.

**SAMPLE INPUT**

2

zazaz

goodarjit

**SAMPLE OUTPUT**

Palindrome

204120000

**Time Limit:**3.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/basic-programming/implementation/basics-of-implementation/practice-problems/algorithm/palindromic-ciphers/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

//string s = "goodarjit";

string s = Console.ReadLine();

char[] rev = s.ToCharArray();

Array.Reverse(rev);

if (s == new string(rev))

{

Console.WriteLine("Palindrome");

}

else

{

long prod = 1;

foreach (char ch in s)

{

if (char.IsLower(ch))

{

//Console.Write((ch - 'a' + 1) + " ");

prod \*= (int)(ch - 'a' + 1);

}

else

{

prod \*= (int)(ch - 'A' + 1);

}

}

Console.WriteLine(prod);

}

}

Console.ReadLine();

}

}

}