## count the vowels

Write a program to count the number of vowels in the given string.

**Input Format**

Input consists of 1 string.

**Constraints**

NA

**Output Format**

Output print the number of Vowels.

**Sample Input 0**

face

**Sample Output 0**

Number of vowels: 2

//SOURCE CODE

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String str=sc.nextLine();

int vowels=0,i;

for(i=0;i<str.length();i++)

{

char vow=str.charAt(i);

if(vow=='a' || vow=='A' || vow=='e' || vow=='E' || vow=='i' || vow=='I' || vow=='o' || vow=='O' || vow=='u' || vow=='U' )

{

vowels++;

}

}

System.out.println("Number of vowels: "+vowels);

}

}

## Palindrome

Write a program to find whether the given string is a palindrome or not without using string library functions.

Note: The string reads the same backward and forward.

**Input Format**

Input consists of 1 string.

**Constraints**

NA

**Output Format**

If the given string is a Palindrome display “Palindrome”, else display “Not a Palindrome”.

**Sample Input 0**

mam

**Sample Output 0**

Palindrome

//SOURCE CODE

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc=new Scanner (System.in);

String str=sc.next();

String str1="";

for (int i=str.length()-1;i>=0;i--){

str1=str1+str.charAt(i);

}

str=str.toLowerCase();

str1=str1.toLowerCase();

if(str.equals(str1)){

System.out.println("Palindrome");

}

else

System.out.println("Not a Palindrome");

}

}

## First Non-repeating Character

Lee conducted a word game for his colleagues. The game is everyone should say a word that should not have any repeating characters in it. If a single character is repeated then the particular person can't continue the game. Lee finds it difficult to disqualify the person from the game since he is not able to find the nonrepetitive character for all the words. Help him to find the winner of the game by writing a program to find the first element which is non -repetitive i.e that element must not be present anywhere else in the string.

**Input Format**

The first line of the input consists of a string.

**Constraints**

NA

**Output Format**

The output displays a character that is non-repetitive. If all the characters in an input string are repetitive, then display All characters are repetitive.

**Sample Input 0**

teeterson

**Sample Output 0**

r

//SOURCE CODE

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String str=sc.nextLine();

int index = -1;

char fnc = ' ';

for (char i : str.toCharArray()) {

if (str.indexOf(i) == str.lastIndexOf(i)) {

fnc = i;

break;

}

else {

index += 1;

}

}

if (index == str.length()-1) {

System.out.println("All the characters are repetitive");

}

else {

System.out.println(fnc);

}

}

}