import java.util.Scanner;

public class CharCounter {

public static int countChar(String str, char ch, int index) {

if (index >= str.length()) return 0;

int count = (str.charAt(index) == ch) ? 1 : 0;

return count + countChar(str, ch, index + 1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String input = sc.nextLine();

char ch = sc.next().charAt(0);

int result = countChar(input, ch, 0);

System.out.println(result);

}

}

import java.util.Scanner;

public class DigitCounter {

public static int countDigits(int num) {

if (num == 0) return 0;

return 1 + countDigits(num / 10);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int number = sc.nextInt();

int digits = (number == 0) ? 1 : countDigits(number);

System.out.println("The number of digits in " + number + " is " + digits);

}

}

import java.util.Scanner;

public class PowerCalculator {

public static int power(int a, int b) {

if (b == 0) return 1;

return a \* power(a, b - 1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int base = sc.nextInt();

int exponent = sc.nextInt();

int result = power(base, exponent);

System.out.println(result);

}

}

import java.util.Scanner;

public class RecursivePalindrome {

public static boolean isPalindrome(String str, int left, int right) {

if (left >= right) return true;

if (str.charAt(left) != str.charAt(right)) return false;

return isPalindrome(str, left + 1, right - 1);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String input = sc.nextLine();

boolean result = isPalindrome(input, 0, input.length() - 1);

System.out.println(result ? "Yes" : "No");

}

}