Buscando rotaciones

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

Un problema clásico e interesante en cualquier lenguaje que permita manejo de cadenas de texto es el de encontrar si una palabra es rotación de otra.

Una palabra A se considera rotación de una palabra B si la palabra A contiene todos los caracteres de la palabra B -sin ningún otro caracter intermedio- pero esta se encuentra girada.

Por ejemplo, atog, toga y ogat son rotaciones de la palabra gato, pero oagt, tago o troga no lo son.

**Entrada**

Leerás dos líneas con una palabra cada una. La primera línea contendrá la palabra A y la segunda la palabra B.

**Salida**

Deberás imprimir “Rotacion” si la palabra A es rotación de la palabra B o “No rotacion” en caso contrario.

**Ejemplo**

|  |  |
| --- | --- |
| gato  toga | Rotacion |
| rota  taro | Rotacion |
| comida  micoda | No rotacion |

*Fuente: Coding Rush 5.2*

Problema subido por: [CodingRush](https://omegaup.com/profile/CodingRush/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Buscando-rotaciones#problems/new-run) | | | | | | | |
| **Envíos** | | | | | | | |

#include <iostream>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

    std::string palabra;

        std::getline(std::cin, palabra);

        std::string rotada;

        std::getline(std::cin, rotada);

    bool estado = false;

    for (int i = 0; i < rotada.length(); i++)

    {

        string post = rotada.substr(i, rotada.length() - i);

        string pre = rotada.substr(0, i);

        if (post + pre == palabra)

        {

            estado = true;

        }

    }

    //Console.WriteLine(estado);

    if(estado) {

        printf("Rotacion**\n**");

    } else {

        printf("No rotacion**\n**");

    }

    getch();

    return 0;

}