

STAR LIGHT



Objectif

On vous donne une rangée de **N** étoiles, représentée par une chaîne de 0 ou 1, avec un total de **N** caractères dans la chaîne.

0 signifie que l'étoile est **éteinte**.

1 signifie que l'étoile est **allumée**.

Le caractère le plus à gauche est l'étoile 1.

Le caractère le plus à droite est l'étoile **N**.

Chaque étoile a un interrupteur indépendant qui vous permet de l'allumer ou de l'éteindre.

Pour allumer ou éteindre une étoile, il existe deux règles :

Règle 1 : Vous pouvez changer l'état de l'étoile i si $i+1$ est allumée ET, si $i < n - 1$, $i+2$, $i+3$,... N sont éteintes.

Règle 2 : La règle 1 ne s'applique pas à l'étoile N , qui peut être allumée ou éteinte à volonté.

Le jeu commence avec un schéma d'éclairage donné **start**. Vous aurez également un modèle d'éclairage cible **target**.

Trouvez le nombre minimum d'interrupteurs nécessaires pour changer le schéma de **start** à **target**.

Paramètres d'entrée et sortie

Entrées

Ligne 1 : Modèle **start** - une chaîne de 0 ou 1, comportant au total **N** caractères

Ligne 2 : Modèle **target** - une chaîne de 0 ou 1, comportant au total **N** caractères

Sortie

Ligne 1 : Le nombre minimum de commutations nécessaires pour changer le schéma d'éclairage de **start** à **target**.

Contraintes

$1 \leq \mathbf{N} \leq 25$

Exemples

Pour changer le modèle de 1101 à 0100 :

1101 (start)

1100 (commutateur n° 4, selon règle 2)

0100 (commutateur n° 1, selon règle 1) - a atteint la cible en changeant deux fois.

1101 (entrée start)

0100 (entrée target)

2 (sortie)

101010 (entrée start)

010101 (entrée target)

26 (sortie)

11001001000 (entrée start)

10000110011 (entrée target)

877 (sortie)



How to start

Java

```
import java.util.*;
import java.io.*;
import java.math.*;

class Solution {

    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        String start = in.nextLine();
        String target = in.nextLine();

        System.out.println("answer");
    }
}
```

Javascript

```
const start = readline();
const target = readline();

console.log('answer');
```

PHP

```
<?php
$start = stream_get_line(STDIN, 255 + 1, "\n");
$target = stream_get_line(STDIN, 255 + 1, "\n");

echo("answer\n");
?>
```



C++

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    string start;
    getline(cin, start);
    string target;
    getline(cin, target);

    cout << "answer" << endl;
}
```

Rust

```
use std::io;

macro_rules! parse_input {
    ($x:expr, $t:ident) => ($x.trim().parse:::<$t>().unwrap())
}

fn main() {
    let mut input_line = String::new();
    io::stdin().read_line(&mut input_line).unwrap();
    let start = input_line.trim_end().to_string();
    let mut input_line = String::new();
    io::stdin().read_line(&mut input_line).unwrap();
    let target = input_line.trim_end().to_string();

    println!("answer");
}
```