

Assignment Two - Programming

Due day

May. 10, 2021

Please hand it in **on time**, and the late submission will be punished.

Problem: Locker

A password locker with N ($N \leq 1000$) digits, each digit can be rotated to 0-9 circularly. You can rotate 1-3 consecutive digits up or down in one step. For examples:

```
567890 → 567901 (by rotating the last 3 digits up)
000000 → 000900 (by rotating the 4th digit down)
```

Given the current state and the secret password, what is **the minimum amount of steps** you have to rotate the locker in order to get from current state to the secret password?

Input

Two strings with equal length (≤ 1000) consists of only digits are given, representing the current state and the secret password, respectively.

Output

Output **one integer**, the minimum amount of steps from the current state to the secret password.

Sample

```
Input:111111 222222
Output:2
```

```
Input:896521 183995
Output:12
```

```
Input:0 0
Output:0
```

Test case

```
Input:1234567890 0987654321
```

Input: 15102342 15123094

Note

Keypoints

- Please use **C++** to implement above algorithm and provide screenshots of the output results
- You can use dynamic programming methods, but pay attention to the time and space complexity of the code
- Please make sure there are **necessary comments** in your source code. Plagiarism is strictly forbidden.

Submission

- Compilable C++ source codes
- A introduction documentation (PDF is recommended). The document should include **algorithm idea, algorithm complexity analysis, and screenshots of running results.**
- Pack all above files and compress it into a **ZIP** file. Please rename the ZIP file as **'StudentID_Name_Assignment_2.zip'**
- Send the zip file to the email of TA:
 - **Mon. 3-4** 354207983@qq.com
 - **Mon. 5-6** 792093953@qq.com