



UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y
MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

ACM CHAPTER CUSCO

CONCURSO DE PROGRAMACIÓN

CUSCONTEST TEST

PROBLEMSET CON SOLUCIONES

Cusco, 14 de Julio del 2024

Este problemset contiene 13 problemas etiquetados de la 'A' a la 'M'.

Información General

A menos que se indique lo contrario, las siguientes condiciones son válidas para todos los problemas.

Nombre del programa

1. La solución debe ser enviada en formatos del lenguaje seleccionado. Ejemplo: `codigo.c`, `codigo.cpp`, `codigo.java`, `codigo.py`, `codigo.cs`.

Entrada

1. La entrada debe ser leída desde la entrada estándar (consola).
2. La entrada consiste en un único caso de prueba, que es descrito en el formato de cada problema. No existen datos extras en la entrada.
3. Cuando una línea de datos contiene muchos valores, estos son separados por exactamente un espacio entre ellos. No existen otros espacios en las entradas.
4. Se utiliza el alfabeto inglés. No hay letras con tildes, diéresis, eñes, u otros símbolos.

Salida

1. La salida debe ser escrita como salida estándar (consola).
2. El resultado debe ser escrito en la cantidad de líneas especificada para cada problema. No debe imprimirse otros datos. Ejemplo: no incluir: “ingrese el número”.
3. Cuando una línea de datos de salida contiene muchos valores, estos deben ser separados por exactamente un espacio entre ellos. No deben imprimirse otros espacios en las salidas.
4. Debe ser utilizado el alfabeto Inglés. No letras con tildes, diéresis, eñes, u otros símbolos.

Límite de Tiempo

1. El límite de tiempo informado para cada problema corresponde con el tiempo total permitido para la ejecución completa de los casos de prueba.

Consejos

1. Para leer múltiples números en una línea en Python usa: `A = [int(x) for x in input().split(' ')]`
2. Para soluciones en java, enviar el archivo `.java` sin el “package name”.
3. Para compilar con `c++` y si el archivo se llama `code.cpp` usar el comando `g++ code.cpp -o code` y para ejecutar usar el comando `./code`

Problemas coordinados por Justino Ferro Alvarez, y planteados por:

Autor	Cargo	Institución
Ulises Mendez	Software Engineer	Google, USA
Erick Alvarez	Software Engineer	Google, USA
Rafa Diaz	Canada & LATAM Community Advisor	Google, CA
Grover Castro	PhD in Computer Sc.	Universität Leipzig, DE
Kleiber Ttito	Software Engineer	Amazon, USA
Jared León	PhD Stud. in Maths	University of Warwick, UK
John Vargas	Senior Data Scientist	Topaz, USA
Josué Nina	Data integration and ETL developer	Provista, USA
Justino Ferro	Software Developer	PE

Problema A. Combinacion De La Cerradura

Archivo de entrada:	Entrada estándar
Archivo de salida:	Salida estándar
Límite de tiempo:	2 segundos
Límite de memoria:	64 megabytes
Autor:	Kleiber Ttito

Scrooge McDuck guarda sus ahorros más preciados en una caja fuerte con una combinación de cerradura. Cada vez que quiere poner allí los tesoros que ha ganado de manera justa, tiene que abrir la cerradura.

Figura 1: Cerradura de Scrooge McDuck



La combinación de la cerradura está representada por n discos giratorios con dígitos del 0 al 9 escritos en ellos. Scrooge McDuck tiene que girar algunos discos para que la combinación de dígitos en los discos forme la combinación secreta. Entonces, en un solo movimiento, puede girar un disco un dígito hacia adelante o hacia atrás. En particular, en un movimiento puede pasar del dígito 0 al dígito 9 y viceversa. ¿Qué número mínimo de acciones necesita para eso?

Entrada

La primera línea contiene un único número entero n ($1 \leq n \leq 1000$), el número de discos en la combinación de la cerradura.

La segunda línea contiene una cadena de n dígitos, el estado original de los discos.

La tercera línea contiene una cadena de n dígitos, la combinación de Scrooge McDuck que abre la cerradura.

Salida

Imprime un único número entero, el número mínimo de movimientos que Scrooge McDuck necesita para abrir la cerradura.

Ejemplo

Entrada estándar	Salida estándar
5 82195 64723	13

Explicación

En la muestra necesita 13 movimientos:

- 1 disco: $8 \rightarrow 7 \rightarrow 6$

- 2 disco: $2 \longrightarrow 3 \longrightarrow 4$
- 3 disco: $1 \longrightarrow 0 \longrightarrow 9 \longrightarrow 8 \longrightarrow 7$
- 4 disco: $9 \longrightarrow 0 \longrightarrow 1 \longrightarrow 2$
- 5 disco: $5 \longrightarrow 4 \longrightarrow 3$

Solución:

Conocimientos requeridos: Fuerza Bruta, Implementación.

Para cada símbolo debemos determinar cómo rotar el disco. Esto se puede hacer mediante la fórmula: $\min(\text{abs}(a[i] - b[i]), 10 - \text{abs}(a[i] - b[i]))$ o incluso mediante los dos ciclos 'for': en ambas direcciones.

Implementación en C++:

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int main(){
4     int n, ans = 0;
5     string a, b;
6     cin>>n>>a>>b;
7
8
9     for(int i = 0; i < n; i++){
10         int num1 = a[i] - '0';
11         int num2 = b[i] - '0';
12
13         int aux1 = abs(num1 - num2);
14         int aux2 = 1 + num2 + abs(9 - num1);
15         int aux3 = 1 + num1 + abs(9 - num2);
16         ans += min(aux1, min(aux2, aux3));
17     }
18
19     cout<<ans<<endl;
20     return 0;
21 }
```