

# MasterMind. Comparar códigos. Versión con valores repetidos.

Mi hermano pequeño quiere jugar una partida de MasterMind. Siempre quiere poner él el código para que yo lo adivine. El problema es que se equivoca al decirme los colores que tengo colocados o descolocados, y resulta completamente imposible adivinar el código que ha puesto. Al final acabo dándole por vencido y se pone muy contento. Hoy he decidido que no podemos seguir así y le he dado una aplicación que le calcula automáticamente el número de colocados y descolocados de mi jugada.



## *Requisitos de implementación.*

La función que compara los códigos debe cumplir los requisitos de la práctica 2 de FP. Para ello, se debe declarar un tipo `tColor`, con los posibles colores y un tipo `tCodigo` que represente un vector de colores. El vector se declarará de un tamaño `TAM_CODIGO`. La función recibe dos vectores, uno con el código y otro con la hipótesis y devuelve los valores colocados y descolocados como parámetros.

Para la lectura de los datos se utilizará una función `char2color` que transforme un carácter en un color.

Para comparar los códigos se puede utilizar un vector auxiliar de booleanos donde se anote los elementos que están colocados. Se comenzará viendo que colores están colocados y después, con ayuda del vector auxiliar se comprobará que valores están descolocados.

## Entrada

La entrada estará compuesta por una serie de casos de prueba, cada uno en una línea. Cada línea contiene dos códigos válidos (de longitud 4); el primero es el código oculto mientras que el segundo es la hipótesis lanzada por el jugador que trata de adivinarlo. La entrada termina con una línea con dos X que no debe procesarse.

Las letras posibles son R (rojo), Z (azul), V (verde), A (amarillo), M (marrón) y B (blanco). Las letras del código serán siempre mayúsculas, sin embargo las letras de la hipótesis pueden ser mayúsculas o minúsculas.

## Salida

Por cada caso de prueba se escribirá una línea con dos números indicando la respuesta dada al jugador que lanza la hipótesis. El primero será el número de elementos bien colocados y el segundo el número de elementos mal colocados.

## Entrada de ejemplo

```
BBMM RzVa
BBMM RRBR
BBMM rBrR
BBMM mmmM
X X
```

## Salida de ejemplo

```
0 0
0 1
1 0
2 0
```

**Autor:** Isabel Pita.