

## Mastermind

Mastermind es un juego de mesa para dos jugadores que consiste en intentar descubrir una clave de colores. Uno de ellos (el *codemaker*) crea la clave que debe ser descubierta en un máximo de intentos por el otro jugador (el *codebreaker*). En cada intento de adivinar la clave, el *codebreaker* escribe una posible clave de colores y el *codemaker* le responde indicando cuántos colores están correctos y además en su sitio y cuantos colores están presentes pero no en su sitio. En el juego original había 4 posiciones y 6 colores posibles para cada posición, con posibilidad de repetir colores. Se deberá escribir un programa que permita jugar al Mastermind a dos jugadores, pero para hacerlo más interesante puede haber hasta 100 posiciones y 20 colores representados mediante letras mayúsculas a excepción de la Ñ. El programa deberá indicar, para cada intento de acertar la clave, cuántos colores están en su sitio y cuántos colores están presentes pero no en su sitio. Un ejemplo de partida con 4 posiciones y 6 colores sería el siguiente.

### EJEMPLO

```
4 6 10      <-- 4 posiciones, 6 colores, 10 intentos
A C A B      <-- Clave a adivinar
A B C D      <-- 1º intento de acertar la clave
1 2         <-- Respuesta del programa. La A está en su sitio (1), B y C están, pero no en
su sitio (2)
C D E F      <-- 2º intento de acertar la clave
0 1         <-- Respuesta del programa.
A B A B      <-- 3º intento de acertar la clave
3 0         <-- Respuesta del programa.
A C A C      <-- 4º intento de acertar la clave
3 0         <-- Respuesta del programa.
A B A C      <-- 5º intento de acertar la clave
2 2         <-- Respuesta del programa.
A C A B      <-- 6º intento de acertar la clave
4 0         <-- Respuesta del programa. Fin del programa por acertar la clave.
```

Sin embargo, para evitar confusiones, a partir de ahora pondremos la entrada y salida separadas.

### Entrada

La primera línea de entrada para el programa debe contener, por este orden, el número de posiciones (*np*), número de colores (*nc*) y número máximo de intentos permitidos (*imax*), separados por un espacio. En la segunda línea se introduce la clave, que contendrá *np* letras mayúsculas, separadas por un espacio. A partir de la tercera línea se introducen intentos de adivinar la clave, uno por línea, con el mismo formato que la línea de la clave.

$0 < np \leq 100$

$0 < nc \leq 20$

$A \leq color \leq T$       {A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T}

$0 < imax \leq 100$

### Salida

A partir de la tercera línea de la entrada, se deberán generar respuestas a los intentos de acertar la clave. Las respuestas consistirán en una línea con 2 números enteros que indicarán, el primero, cuántos colores están en su sitio, y el segundo, cuántos colores están presentes pero no están en su sitio. La ejecución se acabará cuando se acierte el código o cuando se llegue al número máximo de intentos *imax*.

### Puntuación

- **Test 1 (20 puntos):**  $0 < np \leq 4$ ,  $0 < nc \leq 6$ . Ni la clave ni los diferentes intentos tendrán colores repetidos. En los diferentes intentos no habrá colores fuera de su sitio, por tanto el segundo entero de salida será siempre 0.
- **Test 2 (30 puntos):**  $0 < np \leq 20$ . Ni la clave ni los diferentes intentos tendrán colores repetidos.
- **Test 3 (50 puntos):** Puede haber colores repetidos.

### Ejemplos

Entrada 1:

```
4 6 10
A C B E
A D B F
D C F E
A F D E
A C B E
```

Salida 1:

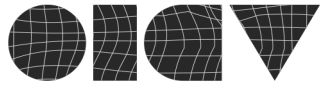
```
2 0
2 0
2 0
4 0
```

Entrada 2:

```
4 6 5
A C B E
```

Salida 2:

```
1 2
0 2
```



Olimpíada Informàtica  
Comunitat Valenciana

## I Olimpíada Informàtica Comunitat Valenciana

Universitat d'Alacant - Escola Politècnica Superior  
Universitat de València - Escola Tècnica Superior d'Enginyeria

```
A B C D
C D E F
A B E F
A C B D
A B C D
```

```
1 2
3 0
1 2
```

Entrada 3:

```
4 6 10
A C A B
A B C D
C D E F
A B A B
A C A C
A B A C
A C A B
```

Salida 3:

```
1 2
0 1
3 0
3 0
2 2
4 0
```