

Práctica 2: Master Mind
20% de la nota de prácticas
Fecha límite de entrega: 8 de enero??

En el juego del *master mind* intervienen dos jugadores. El primero de ellos elige en secreto un número formado por N dígitos distintos. Hay que tener en cuenta que la combinación puede empezar por un cero. El otro jugador trata de adivinarlo mediante tanteos sucesivos. Cada tanteo consiste también en un número de N dígitos (se admiten repetidos), que el primer jugador examina, diciendo cuántos dígitos coinciden y están en su posición correcta (aciertos), y cuántos se han acertado pero no están en su posición (semiaciertos).

En nuestra práctica, vamos a desarrollar este juego con las siguientes restricciones:

- 1) Todos los dígitos de todos los números usados pertenecen al conjunto $\{0, 1, 2, \dots, 9\}$
- 2) El valor de N pertenece al conjunto $\{2, \dots, 6\}$. Su valor se fijará como una constante al principio del programa.
- 3) El número máximo de tanteos que se pueden realizar para acertar el número secreto es 10 .

Modo de Juego 1. Juega un humano.

En este primer modo de juego, será el ordenador el que tenga el rol del jugador que piensa un código. El otro jugador, el que tiene que adivinar la combinación, será la persona que usa el programa.

Para desarrollar este modo de juego será necesario implementar (al menos) las siguientes funciones:

- F1. Escribe una función llamada `codigo_secreto` capaz de generar aleatoriamente un número formado por N dígitos distintos del conjunto $\{0, 1, 2, \dots, 9\}$. Esta función no recibe parámetros de entrada y devuelve como resultado un código secreto válido.
- F2. Escribe una función llamada `aciertos_semiaciertos` tal que, dadas dos secuencias de dígitos, halle el número de aciertos y semiaciertos de la segunda (el tanteo) con respecto a la primera (el código secreto que se trata de adivinar). Por tanto, esta función devuelve dos valores (aciertos, semiaciertos) y recibe como parámetros de entrada el código secreto y el tanteo.

- F3. Escribe una función llamada `valida_tanteo` que compruebe si un tanteo representado como un string, es un tanteo válido. Por ejemplo, los tanteos `'12345'`, `'12'`, y `'3e21'` no son válidos para N igual a 4, mientras que los tanteos `'0123'`, `'9822'` y `'4337'` sí lo son. Esta función recibe como parámetro de entrada un string y devuelve un valor booleano.
- F4. Escribe una función llamada `juega_humano` para simular el primer modo de juego. El proceso será el siguiente: el ordenador piensa un código secreto. Posteriormente y hasta un máximo de 10 oportunidades, se le pide al usuario que introduzca su tanteo. El programa calcula el número de aciertos y semiaciertos y se lo comunica al usuario. Si el número de aciertos es N , el usuario gana. En caso contrario, el programa le pide otro tanteo.

Ejemplo de ejecución

Si el código secreto pensado por el ordenador fuera 6478, una posible ejecución sería:

```
Ya he pensado un código!!!
```

```
Tienes un máximo de 10 intentos para adivinarlo....
```

```
Escribe tu tanteo como un código de 4 cifras: 0000
```

```
Aciertos: 0, Semiaciertos: 0. Te quedan 9 intentos.
```

```
Escribe tu tanteo como un código de 4 cifras: 5854
```

```
Aciertos: 0, Semiaciertos: 2. Te quedan 8 intentos.
```

```
Escribe tu tanteo como un código de 4 cifras: 6378
```

```
Aciertos: 3, Semiaciertos: 0. Te quedan 7 intentos.
```

```
Escribe tu tanteo como un código de 4 cifras: 3478
```

```
Aciertos: 3, Semiaciertos: 0. Te quedan 6 intentos.
```

```
Escribe tu tanteo como un código de 4 cifras: 6478
```

```
Has ganado en 5 intentos!!!
```

Modo de Juego 2. Juega el ordenador

El segundo modo de juego consiste en un programa que adivina la combinación del usuario. El usuario deberá pensar una secuencia y deberá responder, sin mentir, a cada posible solución planteada por el ordenador con el tanteo correspondiente. Una estrategia sencilla, no la mejor, que el programa puede seguir para encontrar la solución se expone a continuación:

El número de combinaciones posibles obviamente es finito y además se puede ordenar fácilmente:

$$0000 \rightarrow 0001 \rightarrow 0002 \rightarrow \dots \rightarrow 9998 \rightarrow 9999$$

La estrategia parte de la primera combinación $C_0 = 0000$, y se pide al usuario una respuesta R_0 con información acerca de los aciertos y semiaciertos. Si el intento no es válido, se tanea al usuario con otra combinación C_1 , pidiendo una nueva respuesta R_1 . De esta forma, tendremos una secuencia de tanteos y respuestas:

$$[(C_0, R_0) \rightarrow (C_1, R_1) \rightarrow \dots \rightarrow (C_i, R_i)]$$

La estrategia para calcular la combinación C_{i+1} es la siguiente: C_{i+1} es el primer código mayor que C_i que cumple $\text{aciertos_semi_aciertos}(C_j, C_{i+1}) = R_j, \forall j = 0, \dots, i$

Para desarrollar este modo de juego será necesario implementar (al menos) las siguientes funciones:

- F5. Escribe una función llamada `siguiente_combinacion` tal que dada una secuencia de tanteos y respuestas, devuelva la siguiente combinación siguiendo la estrategia descrita anteriormente.
- F6. Escribe una función llamada `juega_computador` para simular el segundo modo de juego. El proceso será el siguiente: el usuario piensa un código secreto. Posteriormente y hasta un máximo de 10 oportunidades, la computadora muestra su tanteo y pide al usuario el número de aciertos y semiaciertos. La computadora gana cuando el número de aciertos sea N, y pierde cuando se agote el número de intentos.

Ejemplo de ejecución 1 Suponiendo que el usuario ha pensado el código secreto 3241, la ejecución sería:

```
Mi intento es:0000
Aciertos y semiaciertos: 0,0
Mi intento es:1111
Aciertos y semiaciertos: 1,0
Mi intento es:1222
Aciertos y semiaciertos: 1,1
Mi intento es:3123
Aciertos y semiaciertos: 1,2
```

```
Mi intento es:3214
Aciertos y semiaciertos: 2,2
Mi intento es:3241
Aciertos y semiaciertos: 4,0
He ganado en 6 intentoss!!!!!!!
```

Ejemplo de ejecución 2 Suponiendo que el usuario ha pensado el código secreto 1234, y en caso de que alguna de las respuestas no sea correcta, la ejecución sería:

```
Mi intento es:0000
Aciertos y semiaciertos: 0,0
Mi intento es:1111
Aciertos y semiaciertos: 1,0
Mi intento es:1222
Aciertos y semiaciertos: 2,2
Mi intento es:2122
Aciertos y semiaciertos: 1,1
```

0 me has hecho trampa o alguna respuesta está mal.

Nota

Donald E. Knuth [1] estudió una estrategia para adivinar la secuencia escondida en este juego. In 1994, Kenji Koyama y Tony W. Lai demostraron que dicha estrategia tiene el mejor comportamiento esperado posible. Tampoco en Internet podía faltar información sobre este juego y sus muchas variantes; tómese este punto de partida como ejemplo:

[https://en.wikipedia.org/wiki/Mastermind_\(board_game\)](https://en.wikipedia.org/wiki/Mastermind_(board_game)).

Esta otra también es muy interesante:

<https://www.cut-the-knot.org/ctk/Mastermind.shtml>

References

- [1] D. E. Knuth, The computer as master mind, Journal of Recreational Mathematics 9 (1) (1976) 1–6.