

# Práctica 1

## *El juego de los chinos*

Fecha de entrega: 6 de diciembre

Según una teoría, el conocido como “juego de los chinos” nació en el año 1787 en un pequeño pueblo de León. Felipe Valdeón Triguero, un pastor de la localidad, inventó el juego para pasar las largas horas de pastoreo. Cogió unas pequeñas piedras del suelo e invitó a los demás pastores a que hicieran lo mismo y ocultaran en la mano las que les pareciera bien para adivinar después, con las manos cerradas, cuántas sumaban entre todos.

La forma más tradicional de jugar es que cada jugador dispone de tres monedas y en cada ronda puede sacar 0, 1, 2 o 3 en su mano derecha. Entonces cada jugador, por turno, dice una cifra –su apuesta- intentando adivinar cuántas monedas suman todas las manos, de manera que no se puede repetir una cifra ya dicha por otro. Las reglas de evolución de la partida pueden ser variadas: ‘a la caída’ (el que gana la ronda se retira y se continúan las rondas hasta que sólo queda un jugador), ‘a un marcador de 3 rondas ganadas’ (el que gana 3 rondas se retira y se continúa hasta que sólo queda un jugador), etc. Y una cosa importante... para no confundir a los demás jugadores ¡no se puede mentir! Es decir, un jugador no puede apostar por un número de monedas que sea inviable de acuerdo con las monedas que él ha sacado. Por ejemplo, suponiendo que haya dos jugadores, un jugador debe apostar por un número de monedas que sea mayor o igual que el número de monedas que él ha sacado y menor o igual que el número de monedas que él ha sacado más el número máximo de monedas de que dispone el rival.

### 1. Descripción de la práctica

En esta práctica tienes que ir desarrollando de manera incremental un programa que permita jugar a los chinos. En el juego sólo van a intervenir dos jugadores. Cada jugador elegirá cuántas monedas saca y el ganador de la partida se decidirá en base a ‘marcador de x rondas ganadas’.

**Versión 1 del programa.-** Una vez iniciado el juego, en cada ronda los jugadores (máquina y humano) decidirán cuántas monedas sacan (disponen de 3) y dirán cuántas creen que hay en total (primero la máquina y después el humano). El programa impedirá que el humano mienta en su apuesta -obligándole a decir una cifra viable- así como que repita la cifra dicha por la máquina. El programa decidirá quién gana cada ronda, indicando entonces la

elección hecha por la máquina. El programa deberá permitir jugar todas las rondas que sean necesarias hasta que haya un jugador que consiga 2 rondas ganadas, es decir, vamos a jugar en base a 'marcador de 2 rondas ganadas'.

Una vez determinado el ganador de la partida el programa indicará quién es, así como el número de rondas ganadas por el perdedor.

### Detalles de implementación de la versión 1

Vamos a mantener el marcador de rondas ganadas que determina cuándo acaba la partida en una constante (MARCADOR = 2). Y el número de monedas disponibles en otra constante (NUMMONEDAS = 3). Por último, definiremos otras tres constantes para representar quién gana una ronda: NADIE = 0, para representar que no gana nadie; MAQUINA = 1, para representar que gana la máquina; y HUMANO = 2, para representar que gana el humano.

Y la solución debe incluir la implementación de, al menos, los siguientes subprogramas:

- ✓ **int monedasHumano():** devuelve el número de monedas que saca el jugador humano, quién introduce el valor por teclado. Debe controlar la validez del dato (¡no se pueden sacar más de NUMMONEDAS monedas!) y permitir repetir la introducción del dato por teclado todas las veces que sea necesario hasta que no haya error.
- ✓ **int apuestaHumano(int eleccionM, int numMonedasH):** devuelve el número de monedas elegidas por el humano en la apuesta, controlando que el humano no mienta y que no repita la elección de la máquina. Tiene como parámetros la apuesta de la máquina (eleccionM) y el nº de monedas sacadas por el humano (numMonedasH). Si el humano miente o repite la elección de la máquina el programa le reprenderá y el proceso de introducción de la apuesta se repetirá tantas veces como sea necesario hasta que el humano cumpla las reglas del juego.
- ✓ **int quienGana(int numMonedasM, int numMonedasH, int eleccionM, int eleccionH):** devuelve un valor que indica quién gana la ronda (MAQUINA si gana la máquina; HUMANO si gana el humano) o que nadie ha acertado (NADIE). Tiene como parámetros las monedas sacadas por la máquina y por el humano (numMonedasM y numMonedasH, respectivamente) así como la apuesta de la máquina y la del humano (eleccionM y eleccionH, respectivamente).

Para generar números aleatorios debes utilizar las funciones `rand()` y `srand(semilla)` de la biblioteca `cstdlib`. Una secuencia de números aleatorios comienza en un primer número entero que se denomina semilla. Para establecer la semilla el programa deberá invocar a la función `srand` con el

argumento deseado. Lo que hace que la secuencia se comporte de forma aleatoria es precisamente la semilla. Una semilla habitual es el valor de la hora del sistema que se obtiene con una invocación a `clock()`, de la biblioteca `ctime`, ya que así es siempre distinta para cada ejecución. Así pues, el programa deberá invocar una vez a `srand(clock())`. Una vez establecida la semilla, la función `rand()` genera, de forma pseudoaleatoria, otro entero positivo a partir del anterior. Si quieres que los números aleatorios generados estén en un determinado intervalo, deberás utilizar el operador `%`. Por ejemplo, para obtener un entero aleatorio en el intervalo `[0, limiteSuperior]` hay que usar la expresión `rand() % (limiteSuperior + 1)`; para obtener un entero aleatorio en el intervalo `[1, limiteSuperior]` debe usarse la expresión `1 + rand() % (limiteSuperior)`; y, en general, para obtener un entero aleatorio en el intervalo `[limiteInferior, limiteSuperior]` hay que usar la expresión `limiteInferior + rand() % (limiteSuperior+1-limiteInferior)`.

**Versión 2 del programa.-** En la segunda versión de la práctica hay que ampliar la versión anterior, de forma que el programa permita también configurar tanto el marcador como el número de monedas (que, por defecto, tendrán los valores de la primera versión). Además debe permitir obtener información sobre las reglas del juego, que están almacenadas en el archivo `reglas.txt`. En concreto, el programa debe contar con un menú de opciones como el que figura a continuación y finalizar sólo cuando el usuario decida salir:

- 1 - Cambiar el marcador
- 2 - Cambiar el número de monedas
- 3 - Acerca del juego de los chinos
- 4 - Jugar una partida
  
- 0 - Salir

#### Detalles de implementación de la versión 2

Lo que antes eran constantes, ¿ahora deberán ser variables? Sí, y ello implica adaptar, mediante la inclusión de parámetros, algunos de los subprogramas que hayas implementado (o todos, ¡eso ya lo decides tú!). En todo subprograma que utilizase alguna de las constantes `MARCADOR` o `NUMMONEDAS`, habrá que sustituir el uso de la constante por el correspondiente parámetro. Por ejemplo, la función `int monedasHumano()`; utiliza la constante `NUMMONEDAS`, luego habrá que añadir un parámetro: `int monedasHumano(int numMONEDAS)`.

Además habrá que incluir más subprogramas, al menos las siguientes:

- ✓ **int menu()**: muestra el menú, pide la opción y la devuelve como resultado. Sólo devolverá una opción válida (esto es, en el intervalo `[0,4]`).

- ✓ **int pedirMarcador():** lee de teclado y devuelve un valor de marcador válido.
- ✓ **int pedirMonedas():** lee de teclado y devuelve un número de monedas válido.
- ✓ **bool mostrarReglas():** si el archivo `reglas.txt` no se encuentra devolverá `false`; en otro caso devuelve `true`. Si el archivo se encuentra, mostrará su contenido por pantalla.
- ✓ **void juego(int marcador, int numMONEDAS):** conduce el juego de una partida. Tiene como argumentos el valor del marcador (`marcador`) y el nº de monedas de que dispone cada jugador (`numMONEDAS`).

El archivo `reglas.txt` termina con el centinela `XXX` en la última línea.

**Versión 3 del programa.-** Para que los posibles cambios realizados en el marcador y/o en el número de monedas no se pierdan entre ejecuciones del programa, en esta tercera versión hay que ampliar la versión anterior de manera que se puedan guardar ambos valores en un archivo (`configP1.txt`). Al inicio del programa debe realizarse la carga de los datos desde ese archivo y al final del programa guardar, para la próxima ejecución, los datos posiblemente modificados en el archivo. Si se da el caso de que no se encuentra el archivo al inicio del programa, se tomarán los valores por defecto (los de la versión 1) para el marcador y el número de monedas.

**Versión 4 del programa (opcional: 1 punto del total de la práctica).-** Para finalizar, de manera opcional, puedes extender la versión anterior con otra opción más en el menú que permita jugar una partida de forma automática, es decir, donde la máquina juegue contra sí misma. Ahora el menú de opciones será el siguiente:

- 1 - Cambiar el marcador
- 2 - Cambiar el número de monedas
- 3 - Acerca del juego de los chinos
- 4 - Jugar una partida
- 5 - Jugar una partida automática
  
- 0 - Salir

La solución a esta parte opcional debe incorporar cierta inteligencia por parte de la máquina que elige en segundo lugar.

También puedes, de manera opcional, decidir extender la versión 3 del programa de manera que el jugador humano pueda parar una partida no automática que se esté alargando mucho. Así, cuando se lleve un cierto nº de rondas `MAXRONDAS`, el programa preguntará al humano si quiere continuar

con la partida o quiere abortarla. En caso de decidir abortarla, la ejecución del programa continuará con la visualización del menú de opciones. El valor de MAXRONDAS lo decides tú (por ejemplo, el doble del marcador, el marcador más diez, o lo que consideres oportuno). Si el humano permite que la partida se extienda más de MAXRONDAS rondas, el programa volverá a realizar la misma pregunta cuando hayan vuelto a pasar otras MAXRONDAS rondas (si es que antes no ha acabado la partida), y así sucesivamente.

## **2. Entrega de la práctica**

La práctica se entregará en el Campus Virtual por medio de la tarea **Entrega de la Práctica 1**, que permitirá subir un archivo comprimido conteniendo el archivo con el código fuente de la versión 3 (o de la 4, si haces alguna parte opcional) (archivo `main.cpp`), el archivo de reglas (archivo `reglas.txt`) y el archivo de configuración (archivo `configP1.txt`).

Asegúrate de poner el nombre de los miembros del grupo en un comentario al principio del archivo de código fuente.

**Y recuerda, ¡las prácticas no se copian! Una práctica copiada implica suspender la asignatura.**

Fecha de entrega: **6 de diciembre**

## Ejemplos de ejecución de la versión 1

```

C:\MERCEDDES\Docencia\Curso 2015-2016\Fundamentos_programacion\Practicas\P1\solucion\V1\...
Elige el numero de monedas que sacas <0 a 3>
1
Creo que hemos sacado 5 monedas entre los dos
Ahora es tu turno...
Introduce tu apuesta <0 a 6>
4
La maquina saco 2
El humano saco 1
No ha ganado nadie
-----
Elige el numero de monedas que sacas <0 a 3>
0
Creo que hemos sacado 5 monedas entre los dos
Ahora es tu turno...
Introduce tu apuesta <0 a 6>
2
La maquina saco 2
El humano saco 0
Gana el humano
-----
Elige el numero de monedas que sacas <0 a 3>
1
Creo que hemos sacado 3 monedas entre los dos
Ahora es tu turno...
Introduce tu apuesta <0 a 6>
2
La maquina saco 1
El humano saco 1
Gana el humano
-----
El humano ha ganado la partida!!!
La maquina gano 0 ronda(s)
Presione una tecla para continuar . . . _

```

Ejemplo 1 (partida en la que gana el humano, no hay errores en la introducción de datos)

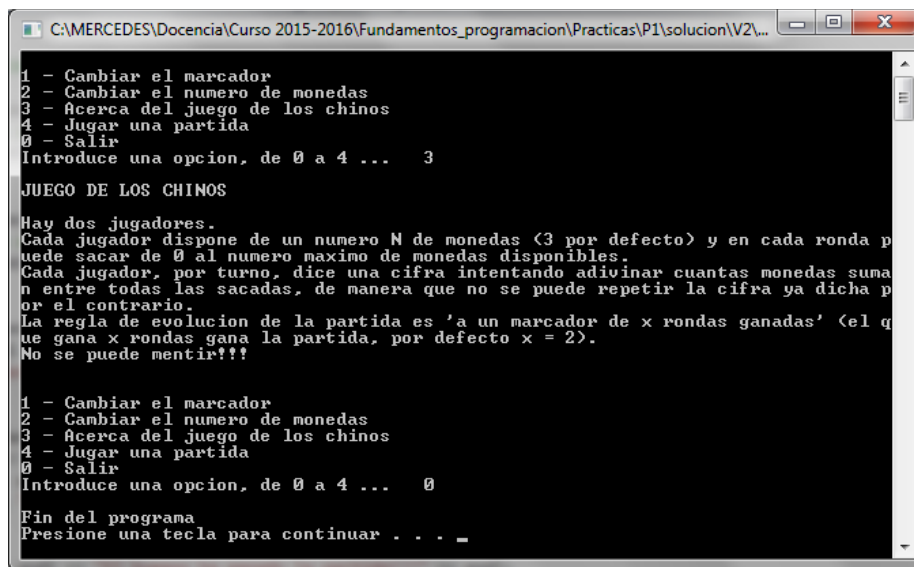
```

C:\MERCEDDES\Docencia\Curso 2015-2016\Fundamentos_programacion\Practicas\P1\solucion\V1\...
Elige el numero de monedas que sacas <0 a 3>
-1
Numero no valido
Elige el numero de monedas que sacas <0 a 3>
4
Numero no valido
Elige el numero de monedas que sacas <0 a 3>
2
Creo que hemos sacado 3 monedas entre los dos
Ahora es tu turno...
Introduce tu apuesta <0 a 6>
1
No mientas... Intentalo de nuevo
Introduce tu apuesta <0 a 6>
-1
Numero no valido
Introduce tu apuesta <0 a 6>
7
Numero no valido
Introduce tu apuesta <0 a 6>
3
No puedes repetir la eleccion de la maquina... Intentalo de nuevo
Introduce tu apuesta <0 a 6>
2
La maquina saco 3
El humano saco 2
No ha ganado nadie
-----
Elige el numero de monedas que sacas <0 a 3>
3
Creo que hemos sacado 3 monedas entre los dos
Ahora es tu turno...
Introduce tu apuesta <0 a 6>
5
La maquina saco 2
El humano saco 3
Gana el humano
-----
Elige el numero de monedas que sacas <0 a 3>
1
Creo que hemos sacado 0 monedas entre los dos
Ahora es tu turno...
Introduce tu apuesta <0 a 6>
1
La maquina saco 0
El humano saco 1
Gana el humano
-----
El humano ha ganado la partida!!!
La maquina gano 0 ronda(s)
Presione una tecla para continuar . . . _

```

Ejemplo 2 (controles de datos varios –datos fuera de rango, mentira por parte del humano, repetición de la apuesta de la máquina por parte del humano, etc.)

**Ejemplos de ejecución de las versiones 2 y 3 (en este último caso asumiendo que el archivo de configuración tiene guardados los valores por defecto para marcador y número de monedas)**



```

C:\MERCEDES\Docencia\Curso 2015-2016\Fundamentos_programacion\Practicas\P1\solucion\V2\...
1 - Cambiar el marcador
2 - Cambiar el numero de monedas
3 - Acerca del juego de los chinos
4 - Jugar una partida
0 - Salir
Introduce una opcion, de 0 a 4 ... 3

JUEGO DE LOS CHINOS

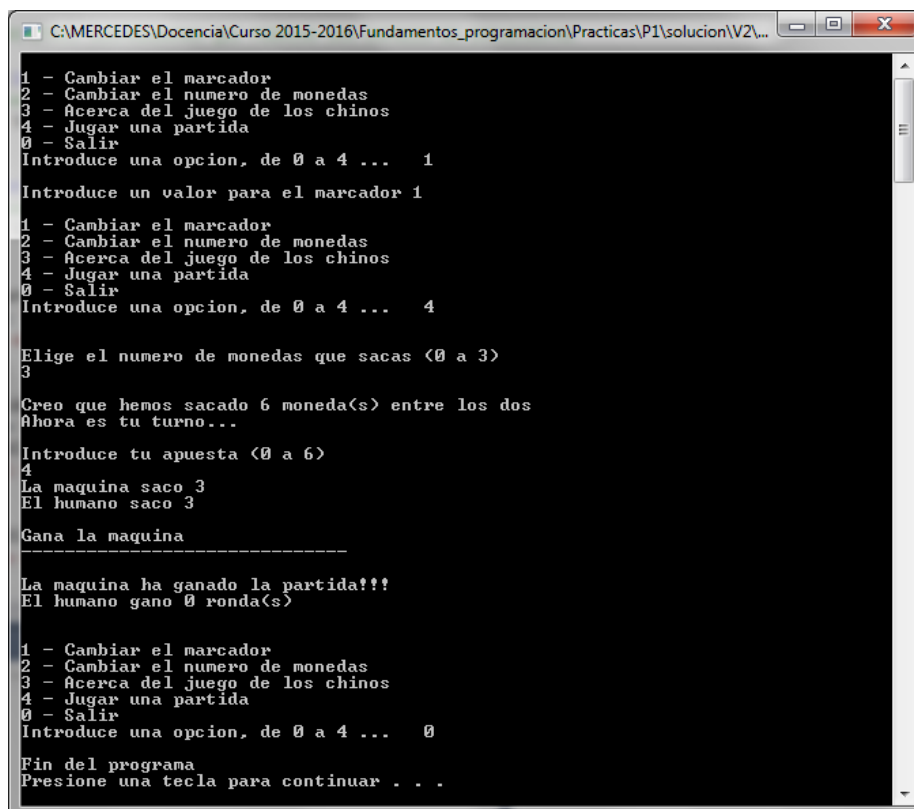
Hay dos jugadores.
Cada jugador dispone de un numero N de monedas (3 por defecto) y en cada ronda p
uede sacar de 0 al numero maximo de monedas disponibles.
Cada jugador, por turno, dice una cifra intentando adivinar cuantas monedas suma
n entre todas las sacadas, de manera que no se puede repetir la cifra ya dicha p
or el contrario.
La regla de evolucion de la partida es 'a un marcador de x rondas ganadas' (el q
ue gana x rondas gana la partida, por defecto x = 2).
No se puede mentir!!!

1 - Cambiar el marcador
2 - Cambiar el numero de monedas
3 - Acerca del juego de los chinos
4 - Jugar una partida
0 - Salir
Introduce una opcion, de 0 a 4 ... 0

Fin del programa
Presione una tecla para continuar . . . _

```

Ejemplo 1 (muestra del fichero de reglas)



```

C:\MERCEDES\Docencia\Curso 2015-2016\Fundamentos_programacion\Practicas\P1\solucion\V2\...
1 - Cambiar el marcador
2 - Cambiar el numero de monedas
3 - Acerca del juego de los chinos
4 - Jugar una partida
0 - Salir
Introduce una opcion, de 0 a 4 ... 1

Introduce un valor para el marcador 1

1 - Cambiar el marcador
2 - Cambiar el numero de monedas
3 - Acerca del juego de los chinos
4 - Jugar una partida
0 - Salir
Introduce una opcion, de 0 a 4 ... 4

Elige el numero de monedas que sacas (0 a 3)
3

Creo que hemos sacado 6 moneda(s) entre los dos
Ahora es tu turno...

Introduce tu apuesta (0 a 6)
4
La maquina saco 3
El humano saco 3

Gana la maquina
-----
La maquina ha ganado la partida!!!
El humano gano 0 ronda(s)

1 - Cambiar el marcador
2 - Cambiar el numero de monedas
3 - Acerca del juego de los chinos
4 - Jugar una partida
0 - Salir
Introduce una opcion, de 0 a 4 ... 0

Fin del programa
Presione una tecla para continuar . . .

```

Ejemplo 2 (cambio del marcador a 1 y juego de partida)

## Ejemplo de ejecución de la versión 4 (incorporando partida automática y asumiendo valores por defecto para marcador y número de monedas en el archivo de configuración)



```
C:\MERCEDES\Docencia\Curso 2015-2016\Fundamentos_programacion\Practicas\P1\solucion\V3...
1 - Cambiar el marcador
2 - Cambiar el numero de monedas
3 - Acerca del juego de los chinos
4 - Jugar una partida
5 - Jugar una partida automatica
0 - Salir
Introduce una opcion, de 0 a 5 ... 5

La maquina 1 saco 3
La maquina 2 saco 2

HABLA LA MAQUINA 1 ...

Creo que hemos sacado 5 moneda(s) entre los dos
Gana la maquina 1
-----
Presione una tecla para continuar . . .

La maquina 1 saco 3
La maquina 2 saco 0

HABLA LA MAQUINA 1 ...

Creo que hemos sacado 4 moneda(s) entre los dos
Ahora es el turno de la maquina 2...
HABLA LA MAQUINA 2 ...

Creo que hemos sacado 1 moneda(s) entre los dos
No ha ganado nadie
-----
Presione una tecla para continuar . . .

La maquina 1 saco 0
La maquina 2 saco 0

HABLA LA MAQUINA 1 ...

Creo que hemos sacado 2 moneda(s) entre los dos
Ahora es el turno de la maquina 2...
HABLA LA MAQUINA 2 ...

Creo que hemos sacado 0 moneda(s) entre los dos
Gana la maquina 2
-----
Presione una tecla para continuar . . .

La maquina 1 saco 0
La maquina 2 saco 2

HABLA LA MAQUINA 1 ...

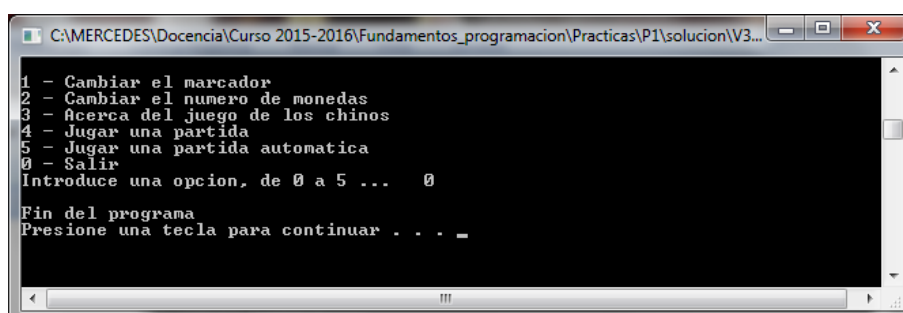
Creo que hemos sacado 1 moneda(s) entre los dos
Ahora es el turno de la maquina 2...
HABLA LA MAQUINA 2 ...

Creo que hemos sacado 2 moneda(s) entre los dos
Gana la maquina 2
-----
Presione una tecla para continuar . . .

La maquina 2 ha ganado la partida!!!
La maquina 1 gano 1 ronda(s)
```

Ejemplo (partida automática)

... continúa el ejemplo



```
C:\MERCEDES\Docencia\Curso 2015-2016\Fundamentos_programacion\Practicas\P1\solucion\V3...
1 - Cambiar el marcador
2 - Cambiar el numero de monedas
3 - Acerca del juego de los chinos
4 - Jugar una partida
5 - Jugar una partida automatica
0 - Salir
Introduce una opcion, de 0 a 5 ... 0

Fin del programa
Presione una tecla para continuar . . .
```