1.– **Juego de los Palillos**: Tenemos k filas con n_k, palillos cada una y dos jugadores que, alternativamente, pueden quitar de una y solo una fila tantos palillos como desee (por lo menos uno), ganando el que se lleva el último. Construya un algoritmo que evalúe el juego, para una posición del mismo dada, indicando la jugada a realizar.

2.– **Juego**: Tenemos N montones cada uno con n_i cerillas ($1 \leq i \leq N$). Dos jugadores cogen alternativamente tantas cerillas como se desee (al menos una), pero de un solo montón. Gana el que se lleva la última cerilla. Construya un algoritmo que evalúe el juego, para una posición del mismo dada, indicando la jugada a realizar.

3.– **Juego**: Tenemos N cerillas ($N > 2$) y dos jugadores. El primero puede iniciar el juego cogiendo tantas cerillas como desee, pero por lo menos una y dejar otra. Van jugando alternativamente, pudiendo cada jugador coger tantas cerillas como quiera pero cogiendo siempre alguna y menos que el doble de las tomadas en la jugada anterior por su adversario. Gana el que se lleva la última cerilla. Construya un algoritmo que evalúe el juego, para una posición del mismo dada, indicando la jugada a realizar.

4.– **Juego**. Tenemos N baldosas cada una con un tamaño b_i ($1 \leq i \leq N$), y un camino para cubrir M metros. Dos personas juegan alternativamente a poner una baldosa de entre las dadas (y no utilizadas aún) hasta cubrir el camino. Pierde aquel que no pueda colocar una baldosa en su turno. Construya un algoritmo que evalúe el juego, para una posición del mismo dada, indicando la jugada a realizar.

Construya también un algoritmo que simule el juego entre una persona y la máquina, suponiendo que la máquina sigue la estrategia utilizada en el párrafo anterior.

5.– **Juego**. Sobre la mesa tenemos un montón de N monedas, cada una de un determinado valor. Dos jugadores cogen alternativamente una moneda del montón. Gana el primer jugador cuyas monedas suman **exactamente** una cantidad dada K, prefijada de antemano. Construya un algoritmo que dada una configuración del juego la clasifique, y devuelva la decisión que produce el resultado indicado. **NOTA**: Puede haber varias monedas con el mismo valor.

Por ejemplo: Si el Montón inicial de monedas es (representadas por su valor) $C = \{1,2,2,3,4,5,6,6,7\}$ y $K = 15$, el desarrollo del juego puede ser como sigue:

- A elige 5 y queda $C = \{1,2,2,3,4,6,6,7\}$ $S_a= 5$ $S_b= 0$
- B elige 7 y queda $C = \{1,2,2,3,4,6,6\}$ $S_a= 5$ $S_b= 7$
- A elige 6 y queda $C = \{1,2,2,3,4,6\}$ $S_a= 11$ $S_b= 7$
- B elige 6 y queda $C = \{1,2,2,3,4\}$ $S_a= 11$ $S_b= 13$
- A elige 4 y **Gana**, puesto que la suma de las monedas que ha elegido *es* $S_a=15$ $(=5+6+4)$

6.– **Juego**. Tenemos un tablero de *M x N* casillas, en cada una de las cuales hay una cantidad de monedas, no necesariamente la misma en cada casilla, y además algunas casillas están vacías. Dos jugadores juegan a ver quien consigue el mayor número de monedas.

El desarrollo del juego es como sigue: Un jugador situado en una casilla *C*, puede desplazarse a una de las casillas adyacentes que tenga monedas y toma todas las monedas que hay en dicha casilla, y a continuación, pasa el turno al otro jugador; si no puede desplazarse a ninguna casilla, pasa el turno al otro jugador. Se supone que el estado inicial del juego es que los dos jugadores están ya posicionados sobre el tablero y las casillas sobre las que están no tienen monedas. Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

7.– **Juego**. Tenemos un tablero de $M \times N$ casillas, en alguna de las cuales hay obstáculos. Dos jugadores juegan alternativamente al siguiente juego. Un jugador situado en una casilla puede desplazarse a cualquiera de las casillas adyacentes, tanto en diagonal, horizontal como verticalmente, que no tenga obstáculos y no haya pasado por ella anteriormente ninguno de los dos jugadores. Si un jugador no puede moverse pasa el turno. Gana el jugador que visita más casillas. Construya un algoritmo que evalúe el juego, para una posición del mismo dada, indicando la jugada a realizar.

8.– **Juego**. Sobre una mesa tenemos una colección de *N* objetos, cada uno de los cuales tiene un peso y valor dados. Dos jugadores cogen <u>alternativamente</u> un objeto de la colección. El peso total de los objetos que coge un jugador no puede sobrepasar una cantidad dada *K*, prefijada de antemano. Gana el jugador que logra reunir la colección de objetos de mayor valor. Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

9.– **Juego.** Tenemos un conjunto de *N* fichas de colores, cada una de un determinado valor, de entre *K* valores posibles ($K<N$). Dos jugadores cogen alternativamente fichas del conjunto, <u>ganando el que coge la última ficha</u>. En cada jugada un jugador puede coger bien todas las fichas de un color, o bien todas las fichas del mismo valor. <u>Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar</u>.

10.– **Juego**. Tenemos un conjunto de *N* enteros positivos. Dos jugadores juegan alternativamente a elegir un número del conjunto y eliminar todos sus divisores. Gana el que elimina el último número. Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

11.– **Juego.** Tenemos un conjunto de *N* enteros positivos. Dos jugadores eligen alternativamente un entero del conjunto y lo eliminan junto con todos sus divisores. Pierde el jugador que elimina el último número. Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

12.– **Juego**. Se considera un número entero de *N* dígitos. Dos jugadores juegan alternativamente. En su turno, cada jugador puede efectuar dos tipos de movimientos:

a) Cambiar un dígito por otro menor. P.e. en el número 9735478 podemos cambiar 5 por 4, 3, 2, 1 ó 0. En concreto, si cambiamos 5 por 2 quedaría 9732478.

b) Si uno de los dígitos es 0, se puede eliminar, eliminando al mismo tiempo todos los dígitos situados a su derecha. P.e., si en número inicial 85972046 elimina el 0, queda el número 85972.

Pierde el jugador que elimina el último dígito.

Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

13.– **Juego**. Se considera una secuencia de *N* dígitos. Dos jugadores juegan alternativamente. En su turno, cada jugador puede efectuar dos tipos de movimientos:

a) Cambiar un dígito impar por otro menor. P.e. en la secuencia 9735478 podemos cambiar 5 por 4, 3, 2, 1 ó 0. En concreto, si cambiamos 5 por 2 quedaría 9732478.

b) Si uno de los dígitos es par ó 0, se puede eliminar, eliminando al mismo tiempo todos los dígitos situados a su izquierda. P.e., si en la secuencia 85972046 eliminamos el 2, queda la secuencia 046.

Pierde el jugador que elimina el último dígito.

Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

14.– **Juego**. Se considera un tablero cuadrado de dimensión *N*. Dos jugadores juegan alternativamente a poner fichas de dimensiones 2×1 en el tablero. Cada uno dispone de $N^2/2$ fichas. En su turno, cada jugador pone una de sus fichas ocupando dos casillas contiguas libres. Uno de los jugadores, por ejemplo el jugador A, puede poner las fichas sólo en horizontal, y el otro jugador, p.e. jugador B, sólo puede poner fichas en vertical. Pierde el jugador que no puede poner ficha en su turno.

Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

15.– **Juego**. La *Devoradora* del tío Antonio aprende rápido, y tras varios días de competir por la comida con la *Listilla*, ha cambiado de estrategia y ahora las dos se han convertido en jugadoras "*de la leche*", y buscan como único fin comer más que la otra, de hecho las dos puede que utilicen la misma estrategia. El juego consiste en que como antes cada una en su turno elige un cubo de uno de los extremos, se lo come y es retirado, ganando la que come más. Diseñe un algoritmo que para una posición del mismo evalúe el juego e indique la jugada a realizar.

16.– **El juego de las reinas**. Darío, “el presi”, entre sus múltiples actividades sociales dentro de la prisión, se ha inventado un nuevo juego con el que está intentando ganar vales de tabaco a sus compañeros de reclusión. El juego que Darío se ha inventado se basa en el problema de las *n* damas de ajedrez que deben ser colocadas en un tablero de *n x n* celdas sin que se ataquen entre ellas, sabiendo que la dama del ajedrez ataca a todas las piezas que se encuentren en su misma fila, columna o alguna de las dos diagonales que pasan por la casilla en la que se encuentra la dama.

En el juego cada jugador, va colocando una dama, y sólo una, en su turno, siempre que la dama colocada no ataque a ninguna de las damas anteriormente colocadas. Aquel jugador que no puede colocar una dama, pierde el juego. Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

17.– **Juego. Dominó, DOS jugadores**. Se considera el juego del dominó con dos jugadores a los que previamente se les han distribuido todas las fichas. Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

18.–**Juego**. Los *Pintados* y los *Dibujados* están pugnando por conseguir el mayor número de escaños en las próximas elecciones comarcales. Los estrategas electorales de cada grupo han recopilado datos sobre cada uno de los pueblos de la comarca y han estimado los votos que son capaces de conseguir según los mítines que van a dar sus máximos dirigentes (solo darán mítines los máximos dirigentes, uno por cada partido) en cada uno de los pueblos en campaña electoral.

Así, para cada pueblo disponen del poder de convicción, en número de votos, de cada uno de sus dos máximos dirigentes. Han realizado la distinción para cada pueblo en si son los primeros en dar el mitin o son los segundos, y también han determinado que un mismo candidato no va a dar dos mítines en un pueblo, porque entonces los electores se “saturan” y “pasan” de votar a dicho candidato.

Puesto que los dirigentes sufren un gran desgaste en cada uno de los mítines que dan, solo van a dar un mitin cada día, y además, van a tomarse un día de descanso entre cada mitin. Así, han suscrito un documento, remitido a la Junta Electoral Central en el que se comprometen a dar mítines en días alternos: un día dará un mitin un candidato y al siguiente lo dará el otro candidato.

<u>Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar. Gana el que consiga más votos</u>.

19.– **Juego. Cuatro en Raya**. Disponemos de un tablero vertical de dimensiones *N x N*, Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

Por ejemplo, si tenemos la posición

o				x
x	x			o
x	o	o	x	o

Posibles continuaciones son:

o	x			x
x	x			o
x	o	o	x	o

O bien

o				x
x	x		x	o
x	o	o	x	o

NOTA: En este caso se trata de ejemplos de jugadas, no de la respuesta que debiera dar el algoritmo desarrollado.

20.– **Juego.** *Jug Ador* y *Lud Opata* han inventado un nuevo juego con el que pasar entretenidos las tardes de verano. El juego se juega sobre un tablero cuadrado de *NxN* casillas en las que inicialmente hay colocadas una serie de fichas de distintos colores. Cada color tiene un valor asignado y el objetivo del juego es conseguir la mayor cantidad de puntos.

El juego se desarrolla de la siguiente manera: *Jug* y *Lud* cogen una ficha alternativamente, de tal manera que al coger una ficha, cogen también las fichas de sus casillas adyacentes. El juego termina cuando no quedan más fichas en el tablero y gana aquel jugador que ha conseguido reunir el mayor número de puntos.

Por ejemplo, dada la posición

az	p	a	na	b
b	m	v	v	a
o	az	na	r	a
o	r	v		z
f	z	p	rs	o

Al coger la ficha de la casilla m tomaríamos las de su alrededor. Si tomáramos la de la casilla rs habría una casilla que no tendría nada, y tomaríamos todas las restantes.

Diseñe un algoritmo que dada una posición del juego, la evalúe, y devuelva la jugada que hace llegar a esa situación.

21.–**Juego GravidBall**. El ***GravidBall*** es un juego de tablero para dos jugadores. En este juego, los dos jugadores juegan alternativamente pulsando uno de los botones del tablero. En cada turno, un jugador tiene la obligación de pulsar un botón y sólo puede pulsar un botón por turno. El tablero de juego es una rejilla de *NxN* casillas en las que hay unas bolas con una puntuación. En cada casilla hay a lo sumo una bola. El tablero tiene alrededor un conjunto de botones, cada uno de los cuales está asignado a una fila o una columna. Cuando un jugador acciona un botón, todas las bolas de esa fila o columna caen a un depósito y se suma la puntuación de las bolas al jugador que pulsó el botón. El juego termina cuando no quedan bolas en el tablero y ganará aquel jugador que consiga la mayor puntuación.

	❶	❷	❸	❹	❺	❻	❼	❽	
❶				⑥	①				❶
❷		⑧							❷
❸			⑤			⑦	⑨		❸
❹	③				⑥			①	❹
❺				⑥					❺
❻		⑩				⑦	②		❻
❼				②					❼
❽							①		❽
	❶	❷	❸	❹	❺	❻	❼	❽	

Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

22.– **Juego**: En un grafo, cada vértice tiene un cierto valor, el botín. Dos ladrones, partiendo del mismo vértice y de forma alternativa, podrán ir a cualquiera de los vértices adyacentes que aún tengan botín. Si alguno no pudiera moverse, por no tener botín ninguno de sus adyacentes, pasaría el turno al otro. El juego acaba cuando ninguno de los jugadores pueda moverse. Ganará el que más dinero consiga.

23.– **Juego**. Se tiene un grafo dirigido valorado, que representa un mapa de carreteras, en el que el peso de los arcos es la distancia entre ciudades. Dos coches salen de una ciudad *A* y deben llegar a otra *B*. Cada vez avanza uno de ellos y atraviesa uno de los tramos (arcos), siempre que no haya pasado ya por él ninguno de los dos coches. Gana el que llegue a *B* recorriendo el menor número de kilómetros.

24.– **Juego de los Buhoneros**. Dos buhoneros se han apostado sus carromatos a ver quien vende más botellas del milagroso bálsamo del Dr. *Bestfit*. La comarca sobre la que han hecho la apuesta tiene un conjunto de aldeas, unidas por caminos angostos y pedregosos que bordean las montañas. Disponen del mapa con las aldeas y los caminos que los comunican, con el coste de desplazamiento entre dos aldeas contiguas, que es el peaje (monetario) a pagar a *ABACO* (*Asociación de BAndidos de la COmarca*). De la misma forma, y teniendo en cuenta la población de cada aldea, y el nivel de convicción que tiene cada uno sobre los vecinos de cada aldea, saben cuantas botellas venderá cada uno en una aldea (suele ser distinta para cada uno).

Por experiencia saben que no deben visitar dos veces la misma aldea, ni siquiera pasar por ella, porque si los aldeanos no perciben una mejoría en sus males, la emprenden a pedradas con el buhonero y su carromato, destrozándolo.

El mecanismo del juego es el que sigue: *Pluma de Águila* (puesto que lleva una en su sombrero) visita una aldea un día, y *Pico de Oro* descansa, y el día siguiente, es *Pico de Oro* el que visita otra aldea, y *Pluma de Águila* descansa, y así sucesivamente, hasta que ambos no tienen posibilidades de ganar más dinero.

Puesto que hay dificultades para saber cuando no deben pasar por una aldea (entre las "*virtudes*" del bálsamo milagroso no se encuentra la de la comunicación telepática), han acordado jugar sobre el mapa de la comarca que tienen, ya que en el mismo se recogen todos los datos necesarios para jugar.

Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

25.– **Juego de los Buhoneros 2.0**. "***El hombre es el único animal que tropieza dos veces en la misma piedra (o con la misma botella)".*** Dos buhoneros se han apostado sus carromatos a ver quien vende más botellas del milagroso bálsamo del Dr. *Bestfit*. La comarca sobre la que han hecho la apuesta tiene un conjunto de aldeas, unidas por caminos angostos y pedregosos que bordean las montañas. Disponen del mapa con las aldeas y los caminos que los comunican, con el coste de desplazamiento entre dos aldeas contiguas, que es el peaje (monetario) a pagar a *ABACO* (*Asociación de BAndidos de la COmarca*). De la misma forma, y teniendo en cuenta la población de cada aldea, y el nivel de convicción que tiene cada uno sobre los vecinos de cada aldea, saben cuantas botellas venderá cada uno en una aldea (suele ser distinta para cada uno).

El sentido común les dice que no deberían pasar dos veces por la misma aldea, ni siquiera pasar por ella, porque si los aldeanos no perciben una mejoría en sus males, deberían emprenderla a pedradas con el buhonero y su carromato, destrozándolo. Pero lo que su experiencia les ha enseñado es que pueden pasar hasta dos veces por una aldea, aunque la segunda vez, deben hacer un descuento del 50% en la mercancía que venden (curiosamente, venden el mismo número de botellas, y a los mismos que la vez anterior, porque "el hombre es el único animal que tropieza...”). Así pues, la segunda vez que pasan por una aldea el beneficio que obtienen es la mitad del de la primera vez.

El mecanismo del juego es el que sigue: *Pluma de Águila* (puesto que lleva una en su sombrero) visita una aldea un día, y *Pico de Oro* descansa, y el día siguiente, es *Pico de Oro* el que visita otra aldea, y *Pluma de Águila* descansa, y así sucesivamente, hasta que ambos no tienen posibilidades de ganar más dinero.

Puesto que hay dificultades para saber cuando no deben pasar por una aldea (entre las "*virtudes*" del bálsamo milagroso no se encuentra la de la comunicación telepática), han acordado jugar sobre el mapa de la comarca que tienen, ya que en el mismo se recogen todos los datos necesarios para jugar.

Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

26.– **Juego.** *Santa* y *Los Magos* se han retado estas navidades a ver quien reparte más ilusión entre los niños (de entre 0 y 500 años). Puesto que no pueden hacerlo en la realidad, puesto que trabajan en fechas distintas, van a hacer una simulación mediante un juego.

En contexto en el que se desarrolla el juego es que tenemos N casas, habitadas cada una por una familia, a cuyos miembros tienen que repartir un conjunto de los juguetes (o cumplir con sus deseos) que han pedido, lo que les proporciona cierta felicidad (a todos los niños de la casa, de entre 0 y 500 años), no necesariamente la misma para todas las casas. Naturalmente, cumplir con los deseos de los niños (o repartirles los juguetes) requiere un esfuerzo. Puesto que *Santa* y *Los Magos* pueden emplear diferentes estrategias (y por tanto tener costes distintos) con los moradores de una casa, la felicidad que consiguen puede ser distinta. Si bien supondremos que siempre emplean la estrategia que proporciona una mayor felicidad para los moradores (si hay que hacer algo, mejor hacerlo bien). Así, se sabe que para una casa c, la felicidad que reparte *Santa* es FS_c, con un coste CS_c, y la felicidad que reparten *Los Magos* es FM_c con un coste CM_c. Además, al principio, *Santa* tiene una fuerza FS y *Los Magos* otra FM, para repartir juguetes.

Aunque se sabe que ni *Santa* ni *Los Magos* tienen ninguna restricción para desplazarse de un lugar a otro, para hacer el juego más interesante, se han impuesto la restricción de que <u>sólo visitarán casas adyacentes a la que se encuentre en un momento dado, según un mapa que muestra las conexiones entre las casas, y que de entre las casas adyacentes, sólo pueden visitar aquellas que ninguno ya haya visitado anteriormente</u>.

El juego se desarrolla considerando que los movimientos entre las casas se realizan de forma alternativa entre los jugadores, esto es, un jugador hace un movimiento (visita una casa), y el otro espera, y a continuación éste realiza un movimiento, y el otro espera, y así sucesivamente. Siempre y cuando un jugador pueda moverse, porque en otro caso, pasa el turno. **<u>Gana el que consigue repartir una mayor felicidad</u>**.

Construya un algoritmo que dada una configuración del juego, la evalúe, y devuelva la decisión que produce el resultado indicado.

27.– **Juego**. Para cada Escuela de Informática se sabe el número de alumnos que finalizan los estudios este año. Varias empresas necesitan personal para desarrollar sus proyectos informáticos y han contratado a dos empresas cazatalentos (ECT) rivales para que en un plazo de a lo sumo K días les proporcionen personal. Las ECT competirán por captar el mayor número de recién titulados para sus clientes.

Saben el porcentaje de alumnos, sobre los recién titulados, que son capaces de captar tras sus contactos con el alumnado de cada escuela. Este porcentaje no es el mismo para cada Escuela, ni tampoco para cada ECT. Por ejemplo, la ECT A puede captar el 80% de una Escuela, en tanto que a la ECT B tiene un 60% de captación en la misma.

Los Directores Generales de las ECT se han propuesto deslumbrar a sus empresas clientes con su trabajo para demostrarles lo engrasada que está su maquinaria de captación de materia gris. Y han planteado la captación como un juego con una serie de reglas, expuestas a sus clientes. Han dibujado sobre un mapa las posiciones de las Escuelas con las relaciones de proximidad entre ellas. Una ECT que está en una Escuela puede ir a otra Escuela adyacente, que no haya sido visitada anteriormente. Ganará el primero que consiga el número de titulados exigido en los K días. Se supone que la captación en cada Escuela tarda un día. Diseñe una función que dada una posición del juego la evalúe y devuelva la decisión a tomar.

28.– **Juego**. Jason (Viernes 13) y Freddy Krugger (Pesadilla en Elm Street), se han retado a ver quién consigue más víctimas. Pero para darle una versión más académica, se habían propuesto realizar sus fechorías en el campus de la Universidad, concretamente en sus Escuelas y Facultades, y como posibles víctimas los pringaos de turno que se quedan hasta altas horas de la noche trabajando en dichos centros (a partir de ahora, el trabajo del pringao, además de poco agradecido, sería altamente "peligroso"...).

Sabían el número N de centros del campus y habían acordado visitar un centro en cada turno, con tal de que no haya sido visitado anteriormente por uno cualquiera de los dos.

Debido a los diferentes métodos empleados con sus víctimas, la "efectividad" de cada uno es distinta para cada Escuela. Así, para el centro i, el número de víctimas de Jasón es J_i, y el de Freddy F_i, no teniendo porqué coincidir.

Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

29.– **Juego**. Jason (Viernes 13) y Freddy Krugger (Pesadilla en Elm Street), se han retado a ver quién consigue más víctimas. Pero para darle una versión más académica, se habían propuesto realizar sus fechorías en el *Campus de la Universidad*, concretamente en sus Escuelas y Facultades, y como posibles víctimas los pringaos de turno que se quedan hasta altas horas de la noche trabajando en dichos centros (a partir de ahora, el trabajo del pringao, además de poco agradecido, sería altamente "peligroso"...).

Sabían el número N de centros del campus y habían acordado visitar un centro en cada turno, con tal de que no haya sido visitado anteriormente por uno cualquiera de los dos.

Debido a los diferentes métodos empleados con sus víctimas, la "efectividad" de cada uno es distinta para cada Escuela. Así, para el centro i, el número de víctimas de Jasón es J_i, y el de Freddy F_i, no teniendo porqué coincidir.

Además, con motivo del esfuerzo para desplazarse entre los centros, su efectividad para cada centro se reduce según van recorriendo más distancia. Así, tenemos que para centro c, la efectividad de Jasón tras haber recorrido una distancia d es $J_c/j(d)$ ($F_c/k(d)$ para Krugger, respectivamente) siendo $j(d)$ la función para Jasón (resp. $k(d)$ para Krugger).

Construya un algoritmo que dada una configuración del juego la evalúe, y devuelva la decisión que produce el resultado indicado.

30.– **Juego entre dos personas.**

Tenemos un tablero cuadrado en el que en determinadas casillas hay fichas. En cada jugada una persona retira una o varias fichas de una columna o de una fila no pudiendo dejar en medio de las fichas que retira otras fichas. Siempre se ha de retirar al menos una ficha. Los jugadores se van alternando y gana quien coge la última ficha.

Construya un procedimiento que dada una determinada posición del juego determine si ésta es ganadora o perdedora, y si es ganadora la decisión que nos lleva a ganar.

Recuérdese que:

– Una posición es **ganadora** si puede dejarle a su contrincante una posición perdedora.

– Una posición es **perdedora** si no es ganadora, esto es, si cualquier posición que puede dejarle a su adversario es ganadora.

Ejemplos:

1) - - -
x x -
- x x

Posición Ganadora: tomamos las fichas de la segunda columna.

2) x x
x x

Posición Perdedora: siempre dejamos una continuación Ganadora.

3) x - -
x - -
- x x

Posición Perdedora: lo mismo que en 2.

4) x x -
- - -
- - -

Posición Ganadora: tomamos todas las fichas.

5) x x x
x - x
x x x

Una **jugada ilegal** es tomar en la fila 1 sólo las fichas de las columnas 1 y 3, pues está entre ellas otra ficha. Por contra sería **legal** tomar en la segunda fila las dos fichas.

31.– **Juego**. Dos jugadores juegan al siguiente juego. Cada jugador tiene un cartón formado por huecos y operadores (suma, resta, multiplicación y división) de forma alternativa. El cartón tiene un número par de huecos *H* y *H-1* operadores. Cada jugador puede ir rellenando los huecos del cartón, de izquierda a derecha, con diferentes números comprendidos entre *1* y

H/2. Además cada jugador no puede jugar dos veces el mismo número. Cuando un jugador coloca un número N comprendido entre *1* y *H/2*, al rival se le coloca de forma automática el valor *(H/2 + 1) – N* en su mismo hueco del cartón.

Gana el juego aquel que tras evaluar el cartón, el resultado de las operaciones (sin tener en cuenta la precedencia) se queda más cercano de un valor V dado.

Diseñe una función que dada una posición del juego la evalúe, y devuelva la decisión a tomar.

Ejemplo:

Cartón:

	+		*		-		*		+		-		/		*		+	

H = 10, V = 25

Valores posibles que se pueden jugar [1-5]

Comienza el jugador 1, y gana el jugador que se queda más cerca de *V = 25*.

El desarrollo del juego podría ser:

J1 -> 3, J2 -> 5, J1 -> 5, J2 -> 1, J1 -> 1, J2 -> 3, J1 -> 2, J2 -> 4, J1 -> 4, J2 -> 4.

3	+	1	*	5	-	5	*	1	+	3	-	2	/	4	*	4	+	2	=18
3	+	5	*	1	-	1	*	5	+	3	-	4	/	2	*	2	+	4	=38

Jugador 1 -> Resultado = 18 -> |25-18| = 7 --------> GANADOR
Jugador 2 -> Resultado = 38 -> |25-38| = 13

Otro ejemplo:

5	+	3	*	1	-	4	*	4	+	2	-	2	/	5	*	3	+	1	=10,6
1	+	3	*	5	-	2	*	2	+	4	-	4	/	5	*	3	+	1	=22,6

Jugador 1 -> Resultado = 10,6 -> |25-10,6| = 14,4
Jugador 2 -> Resultado = 22,6 -> |25-22,6| = 2,4 --------> GANADOR