



HP SCDS

CodeWars 2018 Madrid

GUÍA DE PROBLEMAS

Contenidos

[0] ¡Bienvenidos a CodeWars! – 1 Punto	3
[1] ¡Bienvenidos a CodeWars! (Expert mode) – 2 Puntos	4
[2] ¿Naciste en año impar? ¡Pues no juegas! - 4 Puntos	5
[3] El árbitro no sabe contar – 4 Puntos.....	6
[4] Calienta, que sales - 4 Puntos.....	7
[5] El Villarato - 6 Puntos	8
[6] Los mejores del partido - 6 Puntos.....	9
[7] Piedra, papel, tijeras – 8 Puntos	10
[8] All Star 3 Point Contest – 10 Puntos.....	12
[9] Nos robaron la cartera - 10 Puntos	14
[10] La zona extraña - 16 Puntos.....	15
[11] Seguridad en las cuentas - 17 Puntos	17
[12] El General Manager rata - 23 Puntos.....	18

[0] ¡Bienvenidos a CodeWars! – 1 Punto

Este es un ejercicio obligatorio, sin el cual no se podrá continuar con el resto.

¡Bienvenidos a CodeWars! Vamos a empezar la competición con un regalo.

Debéis desarrollar un programa que muestre por pantalla el siguiente mensaje de bienvenida:

Welcome to CodeWars!

Entrada

No se requiere

Salida

Welcome to CodeWars!

[1] ¡Bienvenidos a CodeWars! (Expert mode) – 2 Puntos

Coged vuestras toallas y secad el sudor de vuestras frentes después del ejercicio anterior.

Este es el primer problema serio ...

Debéis desarrollar un programa que muestre por pantalla un mensaje de bienvenida personalizado para vuestro equipo.

Entrada

Nombre de equipo, por ejemplo:

WarriorsTeam

Salida

Welcome to CodeWars WarriorsTeam

[2] ¿Naciste en año impar? ¡Pues no juegas! - 4 Puntos

Uno de los más conocidos clubes de Baloncesto de Madrid ha resurgido, pero el entrenador nuevo es muy supersticioso con los números. Piensa que los números impares le traen mala suerte, y quiere que durante un partido nunca haya más jugadores nacidos en año impar que en año par en la pista.

Implementad un programa que reciba como parámetro de entrada las cinco fechas de nacimiento, separadas por coma, de los cinco jugadores que están en la pista, y devuelva:

- "You win", si hay más jugadores nacidos en año par.
- "You lose", si hay más jugadores nacidos en año impar.
- "Invalid number of players", si el programa recibe un número de años de nacimiento distinto de 5

Entrada

1980,1983,1983,1981,1986

1986,2000,1993,1982,1986

1982,1997,1987,1982

1982,2001,1973,1982,1986,1980

Salida

You lose

You win

Invalid number of players

Invalid number of players

[3] El árbitro no sabe contar – 4 Puntos

Al finalizar el partido de baloncesto, el árbitro ha recibido un balonazo en la cabeza con la mala suerte de que ha perdido su capacidad para ordenar números. Por tanto, no sabe qué equipo ha ganado el partido porque no sabe qué número de puntos es mayor.

Ayudémosle con un sencillo programa que reciba dos parámetros enteros, y decida cuál de los equipos es el ganador.

Deberá sacar como salida:

- *"Equipo A ganador"*, si el primer parámetro es mayor que el segundo.
- *"Equipo B ganador"*, si el segundo parámetro es mayor que el primero.
- *"Se juega prorroga"*, si los parámetros son iguales.

Entrada

88 90

Salida

Equipo B ganador

[4] Calienta, que sales - 4 Puntos

En baloncesto se permiten un número ilimitado de sustituciones durante el juego. Eso sí, los cambios deben ser notificados al anotador oficial, que será el encargado de dar permiso al jugador para entrar en la cancha. Además, para que la sustitución sea legal, el balón debe estar inmóvil y el reloj detenido. En caso contrario, se pita falta técnica.

Debéis desarrollar un programa que indique si un cambio ha sido legal o si se debe pitar falta técnica, junto con los motivos para pitarla.

El formato de entrada es:

JugadorQueSale/JugadorQueEntra EstadoBalón/EstadoReloj

Donde,

EstadoBalón vale 0 si el balón está en movimiento y 1 si el balón está detenido.
EstadoReloj vale 0 si el reloj no se ha detenido y 1 si el reloj está detenido.

Balón 0 mov 1 p

Tiempo 0 mov 1 p

El formato de salida es:

Sustitución legal: JugadorQueEntra sustituye a JugadorQueSale

o bien

Falta técnica: Balón en movimiento, Reloj no detenido - JugadorQueEntra sigues en el banquillo

Entrada

Paco Fran 0 0,
 Jose Lucas 0 1
 Alicia Lucía 1 0
 Marta Sara 1 1

Salida

Falta técnica: Balón en movimiento, Reloj no detenido - Fran sigues en el banquillo.

Falta técnica: Balón en movimiento, - Lucas sigues en el banquillo

Falta técnica: Reloj no detenido - Lucía sigues en el banquillo

Sustitución legal: Sara sustituye a Marta

[5] El Villarato - 6 Puntos

Un grupo de ciberdelincuentes ha conseguido hackear la base de datos con los resultados deportivos de la actual temporada de la Liga Colegial. Desafortunadamente los servidores que alojaban esta información no disponían de un sistema de copia de seguridad que garantizase la recuperación de los datos originales en caso de pérdida. Desde la organización se solicita la colaboración de los equipos participantes en el torneo HP CodeWars para identificar a los equipos que han sido víctimas del hackeo.

Pero, ¿qué han hecho los ciberdelincuentes? Han modificado la base de datos de resultados de modo que algunos equipos han sufrido modificaciones en los puntos a favor y puntos en contra obtenidos en los 3 primeros partidos de la liga.

Debéis desarrollar un programa que reciba los marcadores de los 3 primeros encuentros, así como los puntos a favor y puntos en contra. Si los valores de puntos para cada equipo no coinciden con la suma obtenida según los marcadores, el equipo ha sido hackeado. En caso contrario, la puntuación es correcta.

La entrada de datos está compuesta por:

1. Los marcadores de los 3 primeros encuentros.
2. Puntos a favor.
3. Puntos en contra.

marcadorPartido1 marcadorPartido2 marcadorPartido3 PuntosAFavor PuntosEnContra

La salida estará formada por:

- Un mensaje indicando si el equipo ha sido hackeado, "Este equipo ha sido hackeado:", o la puntuación es correcta, "Puntuación Correcta".
- La diferencia de puntos a favor.
- La diferencia de puntos en contra.

La diferencia de puntos debe mostrarse junto con un símbolo "+" o un símbolo "-", dependiendo de la diferencia con los puntos reales.

Entrada

50-20 50-20 50-20 150 60
50-20 50-20 50-20 250 100
50-20 50-20 50-20 100 50
50-20 50-20 50-20 150 70

Salida

Puntuación Correcta

Este equipo ha sido hackeado: PF: +100 PC: +40

Este equipo ha sido hackeado: PF: -50 PC: -10

Este equipo ha sido hackeado: PC: +10

[6] Los mejores del partido - 6 Puntos

El entrenador del equipo de baloncesto es un poco nulo con las cuentas, y suele olvidar quiénes son los máximos anotadores y asistentes al final de los partidos

Se pide hacer un programa que lea el número de jugadores seguido de una línea con la información para cada jugador.

El formato de entrada será:

Nº de jugadores a valorar por el entrenador

NombreJugador1;Puntos;Asistencias



Teniendo en cuenta que puede haber un empate entre varios jugadores, el programa deberá:

- Calcular el máximo anotador.
- Calcular el máximo asistente del partido.

El formato de salida será:

El máximo anotador es NombreJugador

El máximo asistente es NombreJugador

Y en caso de empate se mostrarán los mensajes en plural y con los nombres de los jugadores separados por punto y coma (;):

Los máximos anotadores son NombreJugador1;NombreJugador2

Los máximos asistentes son NombreJugador1;NombreJugador2

Entrada

2

Borja:24/12

Alejandro:12/12

Salida

El máximo anotador es Borja

Los máximos asistentes son Borja;Alejandro

[7] Piedra, papel, tijeras – 8 Puntos

¡Empate!, ¡empate!, ¡hay empate!

Y así amigos, con el vozarrón de Navarro nos hemos despertado esta mañana. No, tranquilos, no es el resultado de ningún partido. Como sabéis, ya nos eliminaron la semana pasada, pero seguimos con el espíritu alegre, ¡porque tenemos todavía la mítica porra de temporada!

A principios de cada año, nos apostamos una buena cenorra, que por supuesto, paga el que menos resultados haya acertado ese año.

¡Resulta que tenemos empate! ¡Los hermanos Gasol se la juegan!

Ahora en el momento, se nos ha ocurrido que, para elegir al perdedor, vamos a echar un clásico de "piedra, papel y tijeras".

¿Cómo que no habéis jugado nunca? Ay, ay, ay...

Os recuerdo las reglas:

- Piedra gana a tijeras
- Tijeras ganas a papel
- Papel gana a piedra

Los dos contrincantes, a la de tres, van haciendo los gestos con sus manos del objeto que quieran usar, y el que menos rondas gane, ¡paga la cena!

En realidad, ya hemos jugado las rondas, pero con los nervios del momento, se nos ha olvidado hacer el recuento :(

¿Nos echáis una mano? Tenemos anotados los resultados de las rondas y estamos buscando un programa que nos diga a quien le toca llenarnos el estómago este año.

El formato de la entrada es:

elección jugador 1|elección jugador 2

La salida esperada:

Victorias Pau Gasol: 1

Victorias Marc Gasol: 2

Empates: 2

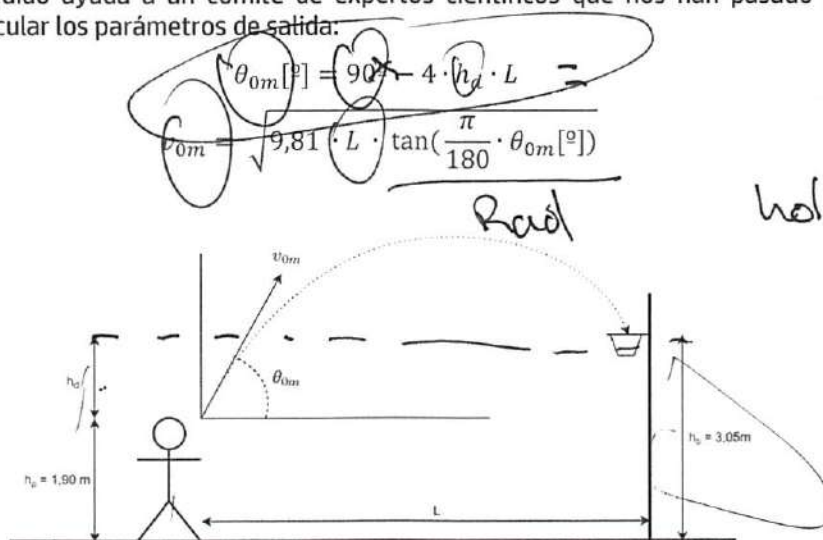
Gracias Pau, ¡todo muy rico!

[8] All Star 3 Point Contest – 10 Puntos

En el All Star de este año se va a realizar una demostración en el concurso de tiros triples, donde un robot de última generación competirá contra los mejores triplistas del mundo.

Nuestra misión será la de crear un programa que calcule la velocidad y ángulo iniciales que se deben emplear para meter canasta desde unas coordenadas introducidas como parámetro de entrada. Los ingenieros mecánicos que han desarrollado el robot han tenido problemas en el suministro de energía, por lo que deberemos buscar la trayectoria para la cuál sea necesario emplear una menor velocidad inicial (consumiendo menos energía).

Para ello, se ha pedido ayuda a un comité de expertos científicos que nos han pasado las siguientes ecuaciones para calcular los parámetros de salida.



La distancia entre el robot y la canasta, designado en las ecuaciones y esquema como L , vendrá dado por la raíz cuadrada de la suma de las coordenadas del robot, introducidas como parámetros de entrada al programa, al cuadrado, es decir:

$$L = \sqrt{x^2 + y^2}$$

Para ciertas operaciones matemáticas, es recomendable usar alguna librería específica para ello: tangente, raíz cuadrada, potencia...

Valor para PI: 3.14159265

$$h_0 = 1.90$$

6 años

[9] Nos robaron la cartera - 10 Puntos

¡Extra! ¡Extra! ¡Oleada de robos en la Comunidad de Madrid! Durante el partido entre Real Madrid y Barcelona correspondiente a la Final de la Copa del Rey de Baloncesto, varios equipos de la Copa Colegial fueron víctimas de diversos robos de material deportivo en sus instalaciones. Ante estos acontecimientos, se ha decidido aumentar la seguridad en los centros deportivos estableciendo contraseñas de acceso a las taquillas.

Cada jugador debe establecer la contraseña de su taquilla, pero debe ser una contraseña segura.

Debéis desarrollar un programa que valide una contraseña según las siguientes reglas:

- Regla 1: La contraseña debe tener una longitud mínima de 8 caracteres.
- Regla 2: La contraseña debe incluir al menos 1 dígito [0-9].
- Regla 3: La contraseña debe incluir al menos una letra minúscula [a-z].
- Regla 4: La contraseña debe incluir al menos una letra mayúscula [A-Z].
- Regla 5: La contraseña debe incluir al menos uno de estos 3 símbolos especiales [% , & , @].

len() < 8

Entrada

soypepu

P4uG4s@l16

Salida

PASSWORD ERROR! Rule 1, Rule 2, Rule 4, Rule 5

PASSWORD OK!

REGEX

[10] La zona extraña - 16 Puntos

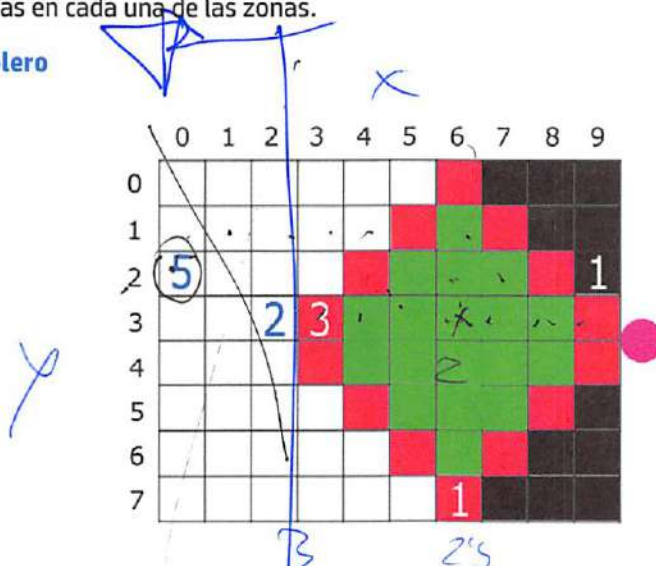
Se ha organizado un torneo de baloncesto en una pista con unas dimensiones y unas zonas de lanzamiento especiales.

Vamos a considerar que:

- La mitad de la cancha de baloncesto es un tablero de 8 filas y 10 columnas.
- Las canastas desde la *zona blanca* tienen un valor con 3 puntos.
- Las canastas desde la *zona roja* tienen un valor de 2.5 puntos.
- Las canastas desde la *zona verde* tienen un valor de 2 puntos.
- Las canastas desde la *zona negra* no suman puntos al marcador.

Debéis desarrollar un programa que permita calcular el número de puntos totales conseguidos según las canastas encestadas en cada una de las zonas.

Ejemplo pista-tablero



Donde el punto de color morado es la canasta y los números que se muestran en el tablero se corresponden con el número de canastas encestadas desde esa *casilla*.

A continuación, la entrada de datos se corresponde con la posición en pista desde donde se ha encestado (fila y columna separadas por dos puntos ":") y el número de canastas encestadas:

Fila:Columna:NúmeroDeCanastas

La salida debe mostrarse de la siguiente forma:

PUNTOS CONSEGUIDOS: TotaldePuntos

Las entradas y salidas que aparecen a continuación se corresponden con los datos representados en el dibujo de la pista-tablero

Entrada

2:0:5 3:2:2 3:3:3 2:9:1 7:6:1

[11] Seguridad en las cuentas - 17 Puntos

Recientemente habéis sido contratados por la ACB para formar parte del equipo de seguridad informática que gestiona las cuentas bancarias de los clubs de esta competición.

Una de vuestras primeras tareas es implementar un sistema de seguridad para garantizar que sólo los usuarios acreditados puedan acceder a la información de estas cuentas, usando un sistema de usuario y contraseña. Dichas contraseñas no se guardarán en texto plano, sino que se guardará un hash de las mismas.

Un hash es el resultado de una función que mapea datos de tamaño arbitrario a datos de tamaño fijo. La ventaja de utilizarlo para guardar contraseñas es que éstas no están almacenadas en formato legible, sino que lo que se almacena es su hash. Cuando los usuarios introducen su contraseña, se calcula su hash y se comprueba si es el mismo que el almacenado, decidiendo así si la contraseña es correcta o no.

El hash utilizado para guardar las contraseñas es la salida de la función *Fowler-Noll-Vo*, en su variante *FNV-1*. Dicho hash se basa en lo siguiente (pseudocódigo):

```
hash = FNV_offset_basis
for each byte_of_data to be hashed
    hash = hash x FNV_prime
    hash = hash XOR byte_of_data
return hash
```

Como podéis ver consiste en inicializar el hash a un número fijo, y luego, para cada byte de los datos, dicho hash se va multiplicando por un número primo y haciendo XOR (OR exclusivo, operador “^” en los lenguajes de programación comunes) con el byte en cuestión. La versión de FNV-1 a utilizar es la de 32 bits, cuyos valores clave son:

- FNV_offset_basis = 2166136261
- FNV_prime = 16777619

Desarrollad un programa que tome por entrada estándar valores de usuario y contraseña separados en líneas diferentes, y produzca como salida valores del tipo usuario:hash_hexadecimal - es común almacenar los hashes como secuencias hexadecimales en vez de números-.

Pista: Utilizad una función incluida en el lenguaje de programación para convertir un número a una cadena hexadecimal, no implementéis vosotros mismos dicha conversión.

Entrada

luis

hola

Salida

luis:bca068e7

[12] El General Manager rata - 23 Puntos

El General Manager de un equipo NBA, que ha comprado recientemente al equipo, cree que las nuevas tecnologías unidas a la estadística son el futuro del Baloncesto y ha decidido expulsar al entrenador. Como reemplazo, nos ha contratado para que hagamos un programa que decida la alineación para cada partido ayudándonos del conocimiento de las habilidades de cada jugador. Las habilidades son las siguientes:

- Tiro cercano
- Tiro lejano
- Pases
- Dribling
- Defensa
- Rebote

*Para cada posición:
Para cada jugador calculo número y otorgo el mayor*

Cada jugador tendrá un valor, entre 0 y 10, para cada habilidad. Usando estos valores, deberemos determinar cuál es la mejor alineación posible decidiendo para cada posición uno de los jugadores. Cada posición tiene asociados unos coeficientes para cada habilidad. Las posiciones y coeficientes asociados son los siguientes:

	POINT GUARD	SHOOTING GUARD	SMALL FORWARD	POWER FORWARD	CENTER
TIRO CERCANO	0	0	0.2	0.4	0.2
TIRO LEJANO	0.2	0.45	0.3	0	0
PASES	0.45	0.15	0	0	0
DRIBLING	0.15	0.35	0.3	0.05	0
DEFENSA	0.2	0.05	0.1	0.25	0.3
REBOTE	0	0	0.1	0.30	0.5

La entrada de datos serán varias cadenas. La primera de todas contendrá el nombre del equipo. La segunda, en un número entero, el número de jugadores N que conforman la plantilla. Después, N líneas con los datos de cada jugador con la siguiente estructura:

`<NombreJugador SkillTiroCercano SkillTiroLejano SkillPase SkillDribling SkillDefensa SkillRebote>`

Donde cada skill será un valor entre 0 y 10 para cada habilidad especificada anteriormente. Nótese los separadores "<" y ">", que también estarán presentes en las cadenas para el nombre del equipo y el número de jugadores en la plantilla. Ejemplo más detallado en la sección **Entrada**.

El programa deberá recoger estos datos, y calcular la mejor plantilla posible eligiendo en el siguiente orden:

- Obtener el mejor Point Guard posible
- Obtener el mejor Shooting Guard posible
- Obtener el mejor Small Forward posible
- Obtener el mejor Power Forward posible
- Obtener el mejor Center posible

Una vez obtenida la alineación, deberá calcular la calificación global del equipo como una media de la calificación de cada jugador en la posición designada, oscilando esta calificación entre 0 y 5 como número

media de esto

$$6 + 3$$

entero (se deberá redondear el valor para que no tenga decimales). Con estos datos, deberá sacar por pantalla el siguiente resultado:

"Alineacion de *teamName*: *PointGuardName* *ShootingGuardName* *SmallForwardName* *PowerForwardName* *CenterName*. Calificacion *teamValue*."

Donde *teamName* será el nombre del equipo, *PointGuardName*, *ShootingGuardName*, etc serán los nombres de los jugadores elegidos para cada posición y *teamValue* la calificación global del equipo.

Entrada

```
<EquipoA>
<7>
<Durant 8 10 8 9 8 7>
<Davis 10 6 7 7 9 10>
<James 9 10 8 9 8 9>
<Cousins 10 4 4 5 9 10>
<Irving 6 9 10 10 8 4>
<Westbrook 5 10 8 10 7 4>
<Love 7 10 6 4 5 8>
```

Salida

Alineacion de EquipoA: Irving Westbrook James Davis Cousins. Calificacion 5.