

INSTRUCCIONES

Si tienes dudas del examen escríbelas en el papel que se te dio y entrégalas al staff que se encuentran en tu laboratorio, tus preguntas se analizarán y entonces se responderán UNICAMENTE con SI, NO y NO SE PUEDE RESPONDER, por lo que formula las preguntas de manera que se contesten con esas respuestas.

Debes responder solo los problemas que te corresponden, no hay puntos extras por contestar otros problemas de otra categoría

Si eres de Primaria tú examen será de KAREL, en las categorías de Secundaria, OMI y Universidad el examen será en C/C++, el examen tiene un valor de 300 puntos en total.

NIVEL	PROBLEMAS QUE DEBEN DE RESOLVER
PRIMARIA	1,2,3
SECUNDARIA	4,5,6
OMI	5,6,7
UNIVERSIDAD	7,8,9

Cuando tengas listo tu problema puedes enviarlo, seleccionando el problema a enviar y pegando tu código en la caja de texto del sistema de envíos

Puedes enviar tu problema en más de una ocasión, en este examen NO HAY PENALIZACION por enviar más de una vez el examen

Se evaluará solo el último envío de cada problema.

Comité Científico Omijal

1 - Temblor (temblor.txt)

Padilla

por: Javier

::Primaria::

La ciudad de Karelajara misteriosamente ha sido víctima de diferentes temblores, por lo que no ha tenido más remedio que llamar a KarMan para que se encargue de cuidar a los ciudadanos y evitar que algún edificio caiga sobre ellos.

KarMan siempre carga con él 2 zumbadores como las armas letales para defender a la ciudad de los desastres naturales y de los malvados villanos que la atemorizan.

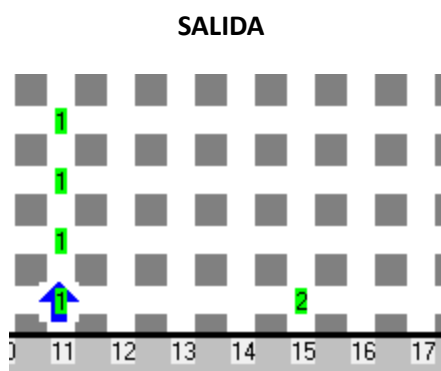
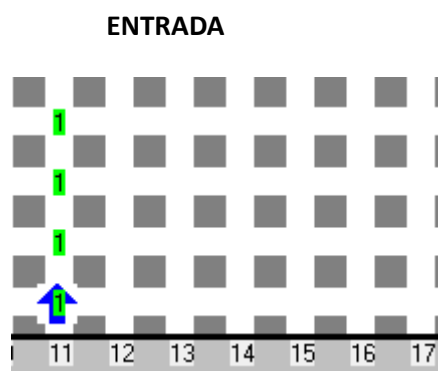
Problema:

Tu misión es ayudar a KarMan a colocar una marca de 2 zumbadores en la primer posición segura donde el edificio no puede caer.

Condiciones:

- Los edificios tienen siempre una altura entre 1 y 99 zumbadores de alto, y están formados por una línea vertical siempre consecutiva de 1 zumbador por posición, su base siempre está junto al suelo (pegado a la pared de la parte inferior).
- KarMan comienza en la base del edificio mirando hacia el norte con 2 zumbadores en su mochila.
- Ten cuidado en tu misión porque El malvado villano Roanigma se acerca...
- No importa la posición, ni la orientación final de KarMan.
- Al terminar el programa el edificio debe estar exactamente igual que al inicio del programa.
- El edificio siempre caerá hacia el lado derecho, su base no se moverá del lugar donde inicia, se desconoce si está pegado a una pared.
- No existen más paredes en el interior del mundo, solo las que delimitan el mundo.

Ejemplo:



2 - Kareltijo (kareltijo.txt)

Padilla

por: Javier

::Primaria::

¡¡Todo se ha descubierto!! El malvado villano JokerGil es que está detrás de los últimos ataques a la ciudad, JokerGil ha capturado a KarMan y encerrado en una cárcel en forma de Caracol, le ha tapado los ojos y escondido en el interior de la prisión.

Afortunadamente has alcanzado a tirar tus dos zumbadores a la entrada y por medio de ellos puedes encontrar una salida.

Problema:

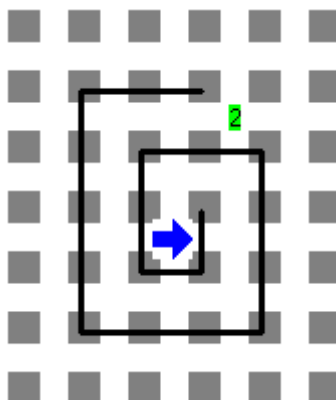
Ayuda a KarMan a salir de esta prisión y encarcelar a JokerGil por sus malvados crímenes.

Condiciones:

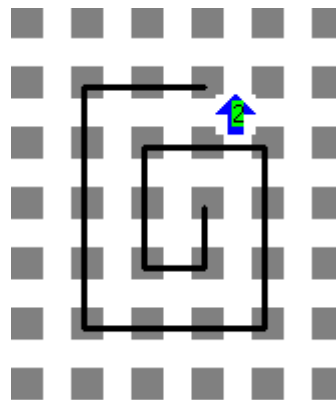
- KarMan comienza en lo más profundo de la prisión en cualquier dirección pero siempre con solo un lado desbloqueado para que pueda escapar.
- La prisión siempre tiene un grosor de una calle.
- No tiene zumbadores en su mochila, y en el mundo solo se encuentran los zumbadores que KarMan tiro a la entrada.
- KarMan DEBERA TERMINAR junto a los dos zumbadores orientado al norte.

Ejemplo:

ENTRADA



SALIDA



3 - Cosechando (cosecha.txt)

García

por: José Luis Roa

::Primaria::

Karelina trabaja en el rancho de su tío Karel cosechando zumbadores frescos, exactamente en marzo es fecha de cosechas así que se dirige a la hortaliza.

PROBLEMA

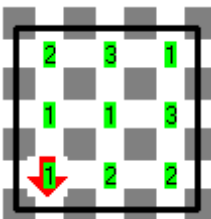
Ayuda a Karelina a recoger únicamente los zumbadores frescos de su hortaliza

CONSIDERACIONES

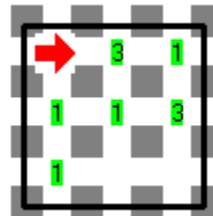
- La hortaliza es de forma cuadrada (cuyo tamaño mínimo es de 3X3).
- Karelina adora comer zumbadores con crema
- Karelina empieza siempre en la esquina inferior izquierda, no se sabe su orientación inicial
- Karelina solo debe recoger los zumbadores frescos que se representan con montones de 2 zumbadores, puede que existan montones de 1 (Verdes) o 3 (Podridos) zumbadores esos montones que no están frescos los debe dejar en su sitio ya que el tío Karel después sabrá que hacer con ellos
- No importa la orientación ni la posición final de Karelina
- Karelina al inicio no tiene zumbadores en su canasta
- Al final Karelina se queda con los zumbadores en su canasta

EJEMPLOS

ENTRADA



SALIDA



4 – Kokepon (kokepon.cpp)

por: José Luis Roa García de la Paz

:: Secundaria :: Tiempo de ejecución 1 seg.

DESCRIPCIÓN

Kokepon es una variante del clásico juego de piedra, papel o tijeras que a Karel le encanta jugar con sus amigos cuando se tiene que decidir algo, en esta ocasión lo tienen que jugar entre Karel y Robocharro para decidir quien escoge la siguiente película en el cine.

Kokepon tiene como diferencia al juego tradicional el identificar a los elementos (Piedra, Tijeras, Papel) con números (1,2,3) respectivamente, respetando las mismas reglas que en el juego tradicional.

PROBLEMA

Realiza un programa que le ayude a Karel a jugar Kokepon con Robocharro

ENTRADA

- Un número N ($1 \leq N \leq 100,000$) que indica la cantidad de juegos que tendrán Karel y Robocharro
- En las siguientes $N+1$ líneas dos pares de números K y C ($1 \leq K, C \leq 3$), donde K es el elemento que selecciono Karel y C el que selecciono Robocharro

SALIDA

- Una línea por juego que indique el nombre del ganador ya sea "KAREL" o "ROBOCHARRO", en caso de empate escribir "EMPATE"

NOTA: Las reglas de quien vence a quien es la siguiente:



EJEMPLO

ENTRADA	SALIDA
4	ROBOCHARRO
1 3	ROBOCHARRO
2 1	EMPATE
3 3	KAREL
1 2	

5 - Botellas (botellas.cpp) por: Gilberto Vargas

:: Secundaria, OMI :: Tiempo de ejecución 1 seg.

DESCRIPCIÓN

Karel organizo una fiesta con jugos, piñatas y pasteles. Al terminar, le toco recoger la casa junto con Robocharro. Como ya estaban muy aburridos, se les ocurrió un nuevo juego.

En una mesa se encontraban N botellas de jugo. En cada turno, cada jugador debe tomar al menos una botella y a lo máximo K botellas. Esas botellas serán puestas en el bote de la basura. El jugador que ponga en la basura la última botella será el perdedor y deberá comprarle una coca y un gansito al ganador.

PROBLEMA

Dadas la cantidad N de botellas en la mesa, y la cantidad de botellas que puede tomar cada jugador, determina quién ganará la partida tomando en cuenta que siempre iniciara Karel.

ENTRADA

La entrada consiste de múltiples casos de prueba. Por cada caso de prueba, será una sola línea con 2 números separados por un espacio, K, y N. La entrada finalizara con un 0.

SALIDA

Por cada caso de prueba una línea, diciendo quien gana, "KAREL" o "ROBOCHARRO", sin comillas.

EJEMPLO

Entrada	Salida
3 15	ROBOCHARRO
4 8	KAREL
2 10	ROBOCHARRO
0	

EVALUACION

Nunca se harán más de 5 preguntas en el mismo caso.

En el 30% de los casos $1 \leq K \leq N \leq 100$

En el 50% de los casos $1 \leq K \leq N \leq 1000$

En el 100% de los casos $1 \leq K \leq N \leq 1,000,000,000$

6 – Genética 3 (genetik.cpp)

por: José Luis Roa García de la Paz

:: Secundaria, OMI :: Tiempo de ejecución 1 seg.

DESCRIPCIÓN

Las investigaciones del Instituto de Genética de Karelstotitlan (IGK) han dado como resultado la manera exacta de cómo saber si un zumbador es pariente directo de otro zumbador, la explicación es muy sencilla. Primero recordemos que la cadena genética de un zumbador está representada por una secuencia finita de N números enteros positivos, teniendo esa lista de números se debe ir comparando por pares adyacentes si el número de la izquierda es mayor al de la derecha en caso afirmativo se sumara uno a un acumulador que se esté manejando, en caso contrario no se le suma nada al final se obtendrá un resultado R. Al aplicar el procedimiento anterior se obtendrán los resultados de los zumbadores que se desean saber si son parientes, lo cual sucede si el resultado de ambos zumbadores es el mismo

PROBLEMA

Realiza un programa que nos diga si dos zumbadores son parientes o no

ENTRADA

- En la primer línea un número N ($2 \leq N \leq 1000$) que nos indica la longitud de las cadenas ADN de los zumbadores
- En la segunda y sucesivas línea la cadena ADN del primer zumbador que está compuesta por una secuencia de N números enteros positivos menores o iguales a 1,000,000 separados entre sí por un único espacio en blanco

SALIDA

- Una única línea que mostrara el carácter 'S' si son parientes o 'N' si no lo son después separado por un espacio el resultado acumulado del zumbador 1 y por otro espacio el resultado acumulado del zumbador 2

EJEMPLO

ENTRADA	SALIDA
10 1 3 5 9 7 10 3 2 10 7 7 5 2 1 0 3 6 7 6 6	N 4 5
ENTRADA	SALIDA
3 10 45 98 1 2 3	S 0 0

7- Rangos (rangos.cpp)

por: Gilberto

Vargas

:: OMI, Universidad :: Tiempo de ejecución 1 seg.

PROBLEMA

La tarea es muy sencilla de hacer pero requerimos de tu destreza para hacerla de una manera óptima. Inicialmente, contamos con una lista de números L y 2 tipos de instrucciones:

- **La instrucción X** recibe 2 números un índice I y un valor K . La tarea de esta instrucción es cambiar el número de la lista L con el índice I por el número K .
- **La instrucción R** recibirá 2 parámetros A y B . Esta instrucción deberá regresar el producto de todos los números de la lista L con los índices entre A y B , incluyendo ambos números. Estos números pueden llegar a ser muy grandes, por lo que tu respuesta deberá ser módulo 1,000,000,009.

Durante la ejecución del programa se te harán una cantidad P de preguntas, donde tendrás que ejecutar cualquiera de las 2 instrucciones descritas.

ENTRADA

- En la primer línea un número N ($1 \leq N \leq 100,000$) con la cantidad de números que contiene la lista (los valores de la lista van desde 1 hasta N).
- En la segunda línea, N números, conformando la lista L .
- En la tercera línea un número P ($1 \leq P \leq 100,000$), indicando la cantidad de preguntas que tendrás que ejecutar.
- En las siguientes P líneas, obtendrás las preguntas con 3 valores distintos, el primero será una letra, que será "X" o "R" indicando que tipo de operación. Los siguientes 2 números serán I ($1 \leq I \leq N$) y K en caso de que la letra sea "X" o A y B (donde $1 \leq A \leq B \leq N$) en caso de que la letra sea "R"

SALIDA

- Por cada pregunta "R" deberás imprimir el resultado de dicha operación en una sola línea.

EJEMPLO NOTA: En todos los casos se hará al menos una pregunta "R".

ENTRADA	SALIDA
10 7 6 1 2 3 4 5 10 9 2 4 R 1 5 X 1 2 X 5 9 R 1 10	252 777600

8 – Salto de rana (salto.cpp)

por: Gilberto Vargas

:: Universidad :: Tiempo de ejecución 1 seg.

PROBLEMA

Una ranita se encontraba en un estanque muy extraño, por lo que se sentía muy incómoda. Decidió irse de ese estanque lo más pronto posible, por lo que te ha pedido que le ayudes a saber cuál es la cantidad mínima de saltos para poder llegar a la salida del estanque.

El estanque está dividido en cuadrados de 1x1 cm y tiene una forma rectangular, para ser más preciso es de N x M cuadritos. Sobre cada cuadrito, se encuentra un hoja-trampolín. Cada hoja-trampolín tiene un número pintado, ese número será la fuerza de la hoja-trampolín. Cada vez que la ranita quiera saltar de una hoja-trampolín a otra solo podrá hacerlo en línea recta, pero saldrá disparado tantas casillas como la hoja-trampolín diga. Por ejemplo, si la ranita esta parada sobre una hoja-trampolín con un numero 3, podrá saltar 3 casillas al norte, al sur, al este o al oeste, pero debes tener cuidado de que la ranita no caiga fuera del estanque. El comienzo del estanque será la hoja-trampolín en la coordenada 1,1 mientras que la salida será la coordenada N,M.

¿Podrás ayudarla?

ENTRADA

- En la primer línea, N y M ($1 < N, M \leq 1000$) separados por un espacio.
- En las siguientes M líneas, N números, con el valor pintado sobre el trampolín.

SALIDA

- Un solo número, la cantidad mínima de saltos para llegar del inicio (0,0) a la salida (N,M).

NOTA:

- Ningún trampolín tendrá una fuerza mayor a 1000.
- Siempre será posible salir del estanque.

EJEMPLO

ENTRADA	SALIDA
5 6 2 3 3 2 1 4 5 3 3 2 3 4 5 3 2 2 1 3 2 3 1 1 1 1 1 2 2 1 1 1	4

9 – Salto de rana 2 (salto2.cpp) por: Gilberto Vargas

:: Universidad :: Tiempo de ejecución 1 seg.

PROBLEMA

Nota: Necesitas leer el problema anterior (Salto de rana) para entender completamente este problema

Una vez que la ranita salió del estanque decidió que no fue tan mala su estancia, así que le gustaría visitarlo una vez a la semana. Como no quiere repetir el mismo camino todas las semanas quiere saber cuántas semanas habrán pasado para que vuelva a repetir un camino suponiendo que siempre tomara uno diferente. En esta ocasión, la ranita solo saltara al sur y al este para no perder tiempo. Ayúdala a saber la cantidad de caminos validos que existen entre el inicio y la salida.

ENTRADA

- En la primer línea N y M ($1 < N, M \leq 1000$) separados por un espacio.
- En las siguientes M líneas, N números, con el valor pintado sobre el trampolín.

SALIDA

- Un solo número, la cantidad de caminos validos entre el inicio (0,0) y la salida (N,M).

NOTA:

- Ningún trampolín tendrá una fuerza mayor a 1000.
- Siempre será posible salir del estanque.

EJEMPLO

ENTRADA	SALIDA
5 6 2 3 3 2 1 4 5 3 3 2 3 4 5 3 2 2 1 3 2 3 1 1 1 1 1 2 2 1 1 1	2