

INSTRUCCIONES

Si tienes dudas del examen envíalas a roa@solacyt.org , o coméntalas en el grupo en Facebook, tus preguntas se analizarán y entonces se publicarán y responderán, ya sea en este sitio (las aclaraciones generales) o en el Grupo de Facebook (las específicas)

Si eres de Primaria los problemas se deben resolver en el lenguaje Karel. Secundaria, OMI, Universidad tú examen de prueba será de C/C++ recuerda que este examen no tiene valor en puntos para tu evaluación final.

<i>NIVEL</i>	<i>PROBLEMAS QUE DEBEN DE RESOLVER</i>
PRIMARIA	1,2,3,
SECUNDARIA, OMI	4,5,6
UNIVERSIDAD	7,8,9,10

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, solo selecciona de nuevo el problema y envíalo

Se evaluará solo el último envío de cada problema.

Puedes enviar tus problemas por email (solacyt@gmail.com) pero SOLO en caso de emergencias, incluyendo tu ID y categoría.

Solo si surgen problemas en el sistema de envíos, se publicaría un MENSAJE en el sitio oficial o en el grupo del Facebook indicando se envíen al email, así que es bueno tengas una copia de respaldo en tu computadora

Tu examen de prueba deberá enviarse antes del lunes 25 de febrero, antes de las 10:01am.

Comité Científico Omijal

1 - Karelstand (karelstand.txt) por: Manuel Fernando Guzmán

::Primaria::

DESCRIPCION

Karel acaba de llegar a Código Ciencia a poner su stand pero el es muy bajito y los letreros de ubicación están en la parte superior y como hay mucha gente , no puede ver donde esta su stand

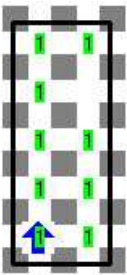
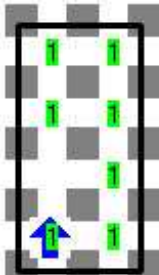
PROBLEMA

Ayuda a Karel a localizar su stand, sabiendo que es el único vacío y solo hay 2 columnas de proyectos.

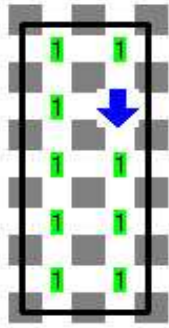
CONSIDERACIONES

- El auditorio será rectangular y se encontrara rodeado por paredes
- Karel se ubicara dentro del auditorio en la primer fila y viendo al norte
- Karel quiere ganar un pase a Expociencias Nacional
- Karel no tiene zumbadores en la mochila
- Los stands ocupados se representan con un zumbador (1) .
- La única posición sin zumbador es el stand de Karel
- Karel deberá detenerse en el lugar vacío (su stand)
- No importa la orientación final de Karel.

EJEMPLO

Ejemplo 1	Ejemplo 2
<p data-bbox="224 1528 367 1556">Entrada 1</p> 	<p data-bbox="820 1528 963 1556">Entrada 2</p> 

Salida 1



Salida 2



2 - Precipicio (precipicio.txt)

por: Fernando Guzmán

::Primaria::

DESCRIPCION

Karel nuestro campeón nacional, frecuentemente se caía de la cama al dormir, pero acaba de comprar en Mercado Libre un sensor contra caídas, y necesita que le ayudes a programarlo.

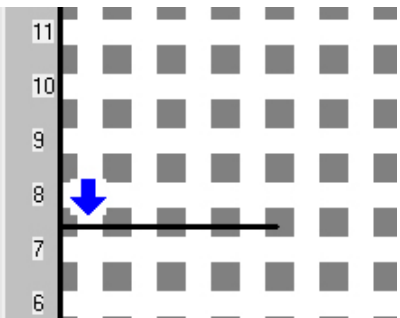
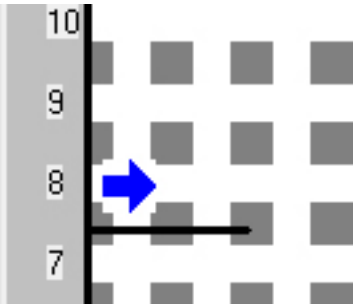
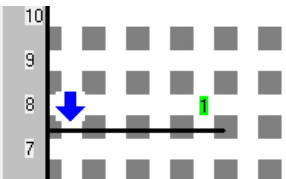
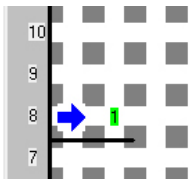
PROBLEMA

Ayuda a Karel a localizar programar su sensor que le permitirá detenerse justo antes de caerse de la cama para dejar ahí su almohada como señal y entonces poder regresar a su posición original y volver a dormir.

CONSIDERACIONES

- Karel se ubicara en la esquina izquierda de su cama sin saber la orientación
- Karel odia caerse de la cama
- Karel inicia con 1 zumbador en su mochila
- Karel deberá detenerse en la última posición de la cama, dejar su almohada y regresar a la posición y orientación original
- La Cama siempre es recta
- La orientación final de Karel debe ser la misma de como inicio.

EJEMPLO

Ejemplo 1	Ejemplo 2
<p data-bbox="224 1262 367 1293">Entrada 1</p> 	<p data-bbox="820 1262 963 1293">Entrada 2</p> 
<p data-bbox="224 1646 342 1677">Salida 1</p> 	<p data-bbox="820 1646 938 1677">Salida 2</p> 

3 – Bache (bache.txt)

por: Manuel Fernando Guzmán

::Primaria::

DESCRIPCIÓN

En karelandia, las lluvias dejaron muchos baches en las calles, lo que ha ocasionado bastantes accidentes, por ello se ha pedido un préstamo al Fondo Internacional para Kareldesastrez y así poder tapar todos los baches de la ciudad.

PROBLEMA

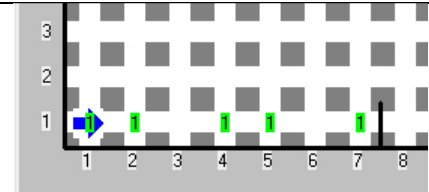
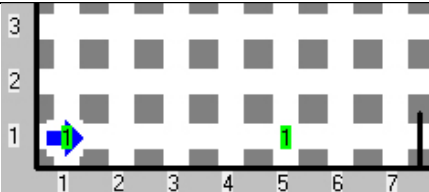
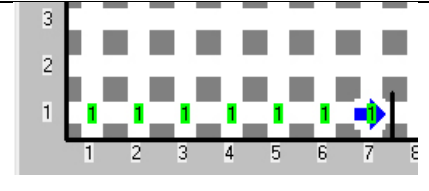
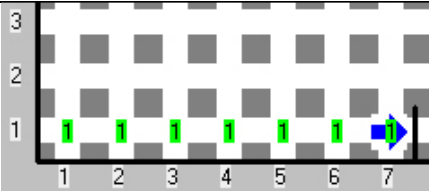
Ayuda a Karel a llenar todos los baches de la ciudad y dejar una calle de primer Karelmundo, deberás llenar solo los lugares vacíos y dejar sin tocar las partes que están en buen estado (que tienen zumbador)

ENTRADA

- Karel inicia en su casa (1,1) orientado al este
- La calle es de solo una posición de alto.
- Karel tiene los suficientes zumbadores en la mochila necesarios
- Sabemos que hay un bache si encontramos una columna vacía
- Al encontrar un bache se debe colocar n zumbador
- El largo de la calle es máximo 100 posiciones
- Se sabe que se llegó al final de la calle al encontrar una pared
- No importa la posición final de Karel

EJEMPLO1

EJEMPLO2

Entrada 1 	Entrada 2 
Salida 1 	Salida 2 

4 –Hasta Cuando (hasta.cpp) por: Fernando Guzmán

:: Secundaria, OMI ::

DESCRIPCIÓN

En tu primer día en secundaria, todos han quedado sorprendidos con tu habilidad para hacer sumas y cálculos, tu respuesta es que has sido olímpico de Omijal desde Primaria y por ello tu nivel es superior, un compañero de clase esta decidido a hacerte quedar mal y te ha dado un reto el cual TU debes superar y demostrar eres el Mejor!

PROBLEMA

Realiza un programa que dados 2 numero al sumarlos, calcules cuantas veces requieres hacer esa suma para poder superar el “numero meta”, por ejemplo si te dan el 5 y el 3 y como meta te dicen el 20, tu sabes que la suma es 8 (5+3) y con ellos puedes calcular que a la tercera suma (24) ya tendrás un numero mayor al “numero meta” y esa es la respuesta correcta (3)

ENTRADA

3 números, donde el primero y el segundo son los números base y el tercero es el “número meta”. Donde ninguno de los números podrá pasar de 10,000

SALIDA

Una sola línea con el numero de sumas necesarias para rebasar el numero meta

EJEMPLO

ENTRADA	SALIDA
3 4 36	6
ENTRADA	SALIDA
1 1 9	5

5 - Buscando (busca.cpp)

por: Fernando Guzmán

:: Secundaria, OMI ::

PROBLEMA

Dado un número entero $1 \leq N \leq 100,000,000$ y un lista de M números enteros $3 \leq M_i \leq 50,000$
Encuentra un conjunto de 3 números de la lista, tales que su suma sea igual al número N .

ENTRADA

Tu programa deberá leer de teclado el numero a buscar N en al primer línea, en la segunda línea el número M y en las siguientes M líneas los números de la lista.

SALIDA

Tu programa deberá obtener como resultado tres números de la lista dada tales que su suma sea igual al número N . Los tres números deberán estar escritos en la misma línea, separados cada uno por un espacio.

NOTA: En el caso de que exista más de una solución, tu programa solo deberá escribir una de ellas.

EJEMPLO

<i>Entrada</i>	<i>Salida</i>
25	4 18 3
10	
15	
8	
1	
4	
18	
3	
32	
12	
4	
10	

6 - Kambiandose (kambiand.cpp) por: José Luis Roa

:: Secundaria, OMI ::

HISTORIA

Karel esta súper emocionado ya que Grettel le dijo que si iba a salir al cine con el, por lo que ahora su problema es escoger la ropa que debe ponerse, más sin embargo no sabe cuantas combinaciones de ropa posible puede realizar.

PROBLEMA

Ayuda a Karel a saber por medio de un programa escrito en C/C++ de cuantas formas puede vestirse de modo que cada conjunto de ropa difiera de los demás en por lo menos un elemento.

ENTRADA

- En la primera línea: C casos que Karel puede tener en el problema, donde $1 \leq C \leq 1'000,000$
- *Línea 2 hasta C+1 (es un caso diferente por línea):* N prendas de ropa que tiene Karel en su armario donde $1 \leq N \leq 100$ y separado por un único espacio el valor R que es el numero de prendas que Karel quiere usar a la vez donde $1 \leq R \leq 100$ y $R \leq N$.

SALIDA

- C líneas indicando en cada una el numero de formas en que se puede vestir Karel, una línea por cada caso expuesto en la entrada

EJEMPLOS

ENTRADA	SALIDA
2	3
3 2	1
10 10	

7 – Fracciones (fracciones.cpp) por: José Luis Roa García de la Paz

::Universidad::

PROBLEMA

Se pide la construcción de un programa capaz de generar ordenadamente todos los números racionales en el intervalo (0, 1) que tengan un denominador entre 1 y un cierto número N. Así, para N = 5, la solución es la serie de racionales 1/5, 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5. Nótese que no se generan números racionales repetidos que se escriban de forma diferente; por ejemplo, la expresión 2/4 no aparece en el resultado porque sí lo hace 1/2, que denota el mismo racional. En estos casos, siempre escribiremos la forma más simplificada posible del racional.

ENTRADA

- Una única línea que contiene uno o dos caracteres que representan un número entero entre 1 y 99, para el que queremos calcular la serie de racionales correspondiente.

SALIDA

- Un número indeterminado de líneas cada una de las cuales es de la forma:

Num Den

donde Num y Den son el numerador y denominador de un racional de la serie correspondiente al número de la entrada, representado cada uno mediante uno o dos caracteres (sin ceros a la izquierda). Entre Num y Den debe aparecer un y sólo un carácter blanco; antes de Num y después de Den no debe aparecer ningún carácter.

EJEMPLO

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

8 – Campo minado (campo.cpp) por: José Luis Roa

::Universidad::

DESCRIPCION

Se nos pide que comprobemos la trayectoria de un robot en un recinto rectangular delimitado por cuatro paredes, cada una de ellas orientada a uno de los puntos cardinales (en sentido horario: N, E, S y O).

El robot está diseñado para desplazarse utilizando una cuadrícula que cubre el suelo del recinto, de modo que puede avanzar en cada paso desde una casilla hasta cualquiera de las que tenga un lado en común con ella. Así, podemos considerar el espacio por el que puede moverse el robot dividido en m filas (numeradas de 1 a m , yendo de N a S) y n columnas (numeradas de 1 a n , yendo de O a E).

El robot se desplaza de acuerdo a tres órdenes diferentes: "giro a la izquierda" (I), "giro a la derecha" (D) y "avance" (A). El avance se realiza de casilla en casilla (una cada vez), pudiéndose sólo pasar desde la casilla en la que se encuentre el robot a cada una de las cuatro (o menos) que tienen un lado en común con esta. En todo momento el robot está orientado directamente hacia uno de los cuatro lados de la casilla en la que se encuentra (que distinguiremos nombrándolos como los puntos cardinales; es decir, en sentido horario: N, E, S y O), y las órdenes I y D hacen girar al robot 90º sin avanzar. De este modo, si por ejemplo el robot tiene orientación N, con la orden I, quedaría en la misma casilla pero con orientación O (pues el giro hacia la izquierda corresponde al giro antihorario); y con la orden D quedaría orientado al E (pues el giro a la derecha es el que se realiza en sentido horario). La orden de avance se aplica siempre en la orientación que tenga en ese momento el robot y ha de tenerse en cuenta que un intento de avanzar contra los límites del recinto provocaría la destrucción del robot.

El problema es que el espacio por el que ha de desplazarse el robot se encuentra minado, de forma que si el robot pisa alguna de las casillas que tiene una mina será destruido. Lo mismo sucede si el robot se estrella contra alguna de las paredes del recinto al recibir una orden de avance incorrecta, como ya hemos dicho.

Los programadores del robot se encargan de generar la secuencia de órdenes que le harían llegar de su posición inicial a un destino. Nuestra responsabilidad es la de supervisar esta secuencia propuesta y asegurarnos que la solución que nos proponen los programadores es correcta y no supone ningún peligro para la integridad del robot.

PROBLEMA

Se trata de comprobar que una secuencia de órdenes determinada desplaza el robot dentro del recinto sin que sea destruido.

ENTRADA

- La primera línea contiene el número de filas (m) y columnas (n) del recinto separados por una coma y sin espacios en blanco. Donde se cumple $1 \leq m < 100$ y $1 \leq n < 100$.
- Las m líneas siguientes (de la 2 a la $m+1$) contienen cada una, una fila del recinto. Concretamente, la línea tendrá n ceros o unos separados por comas y sin espacios en blanco entre ellos. Un 0 corresponde a una casilla que puede ser pisada sin peligro y un 1 corresponde a una casilla minada.
- La siguiente línea (la $m+2$) contiene las coordenadas $(i1, j1)$, $1 \leq i1 \leq m$, $1 \leq j1 \leq n$, de la posición inicial del robot, separadas con una coma y sin espacios en blanco.
- La siguiente línea (la $m+3$) contiene las coordenadas $(i2, j2)$, $1 \leq i2 \leq m$, $1 \leq j2 \leq n$, del destino del robot, separadas con una coma y sin espacios en blanco.
- La siguiente línea (la $m+4$) contiene una única letra indicando la orientación inicial del robot (N, E, S o O).
- La siguiente línea (la $m+5$) contiene el número de órdenes dadas al robot (entre 1 y 40).
- La última línea (la $m+6$) contiene la secuencia de órdenes, es decir, letras "A", "D" e "I", separadas por un único blanco y sin blancos al inicio ni al final de la línea.

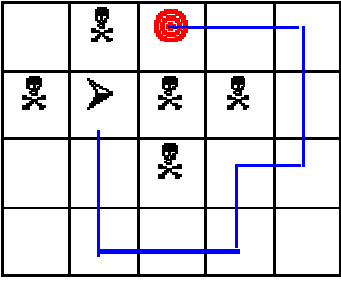
SALIDA

- Una línea que constará de una única letra: una "C", si la secuencia de órdenes lleva de la posición inicial al destino sin destruir el robot, o una "E" en caso contrario.

EJEMPLOS

ENTRADA	SALIDA
4,5 0,1,0,0,0 1,0,1,1,0 0,0,1,0,0 0,0,0,0,0 2,2 1,3 E 16 D A A I A A I A D A I A A I A A	C

N



S

9 – Al revés (alreves.cpp) por: Gilberto Vargas

::Universidad::

DESCRIPCIÓN

Ésta mañana mientras Karel se arreglaba para irse a la escuela, fue encerrado en el mundo de los espejos por su reflejo malvado. Adentro todo estaba totalmente volteado. Como la suerte de Karel fuera del espejo es malísima y dentro del espejo es muy buena, se ha encontrado un papelito donde están las instrucciones para salir del mundo de los espejos. Como todo está al revés, deberás cambiar las mayúsculas por minúsculas y viceversa además de desplegar el mensaje al revés.

PROBLEMA

Ayuda a Karel a traducir el mensaje del papelito.

ENTRADA

Una única línea con el mensaje, solo contendrá letras de la A a la Z (sin la ñ) minúsculas y mayúsculas, y espacios. Donde la longitud de la cadena nunca será mayor a 1,000 posiciones

SALIDA

Una sola línea con el mensaje traducido

EJEMPLO

ENTRADA	SALIDA
IT A ETNERf ATSe ADILAs Al	La Salida Esta Frente a ti

ENTRADA	SALIDA
idnetne aY	yA ENTENDI

10 – Juego de la suerte (juego.cpp) por: Gilberto Vargas

::Libre::

Karel se fue a un concurso de telejuegos a chambear. Como es un fantástico matemático, le han asignado la labor de inventar los juegos. Se le ha ocurrido un juego muy sencillo e interesante. El jugador subirá unas escaleras y tendrá un disco. En cada casilla hay un número que indicara cuánto dinero ganara en esa casilla. Las casillas forman una pirámide, donde el piso de hasta arriba tiene una casilla, el segundo dos, el tercero tres y así sucesivamente, acomodadas como en el siguiente ejemplo (pirámide de 5 pisos):

```
    5
   2 6
  6 7 8
 9 5 3 5
1 2 3 4 5
```

El jugador subirá hasta el último escalón y dejara caer el disco. La pirámide tiene un sensor que detecta cuando el disco pasa por la casilla, así que el disco dejara un rastro de N casillas marcadas. El participante ganara la suma total de las cantidades en las casillas que marque el disco. El disco solo podrá ir hacia abajo, y a una de las 2 casillas adyacentes. Como los de la producción se preocupan por no regalar demasiado dinero, te han pedido que les ayudes a calcular cual es la cantidad máxima de dinero que el concursante puede ganar.

Entrada

En la primer línea, un numero N, que nos dirá cuantos pisos tendrá la pirámide.
De la línea i+1 hasta la N+1, i números, indicando el numero que está en esa casilla.

Salida

Una sola línea con la cantidad máxima que el concursante puede ganar.

Limites

$2 \leq N \leq 1000$

Ejemplo

Entrada	Salida
5 5 2 6 6 7 8 9 5 3 5 1 2 3 4 5	29

