Camino Más Corto

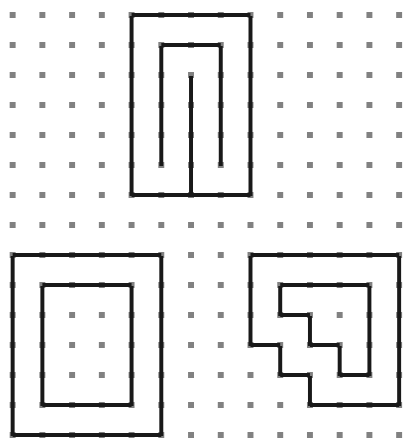
|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Te recomendamos primero leer el problema "Recuerda tu orientación" para poder entender éste.**

**Historia**

Karel está en un circuito. Los circuitos son como los pasillos, pero no tienen ni inicio ni final: ¡si sigues caminando en un circuito, vas a regresar a donde empezaste!

Aquí hay algunos ejemplos de circuitos:



**Problema**

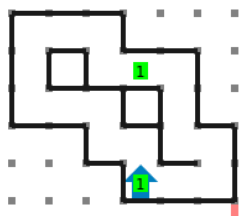
En el circuito hay dos zumbadores. Escribe un programa que encuentre el camino más corto entre los dos zumbadores y lo rellene de zumbadores.

**Consideraciones**

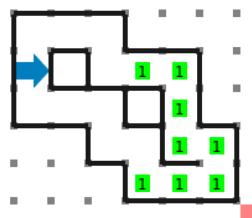
* Karel siempre inicia sobre un zumbador.
* Karel tiene infinitos zumbadores en la mochila.
* Siempre habrán exactamente dos zumbadores en el circuito.
* Karel inicia mirando a la siguiente casilla por la que debe caminar.
* No importa la orientación ni posición final de Karel.
* No pueden quedar zumbadores en el mundo además de los dos iniciales y los que marcan el camino más corto.

**Ejemplo**

**Entrada**



**Salida**



Charcos Cuadrados

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Historia**

A karel le gusta brincar en los charcos. Los charcos se definen como cuadros de zumbadores en un mapa delimitado por paredes. El objetivo de karel es encontrar el charco más grande que hay en el mapa.

**Problema**

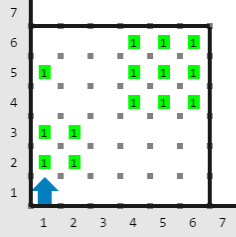
Escribe un programa que le ayude a Karel a encontrar el charco mas grande y al final poner en la esquina inferior izquierda del mundo el número de zumbadores del charco más grande encontrado.

**Consideraciones**

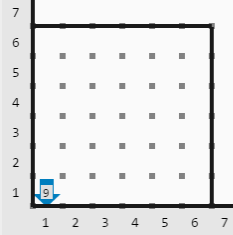
* Karel siempre inicia en la esquina inferior izquierda orientado al norte.
* Karel tiene infinitos zumbadores en la mochila.
* Todos los charcos en el mapa son cuadrados, es decir miden lo mismo de alto que de ancho.
* Los charcos siempre tienen una separacion de almenos una casilla con los demas charcos.
* No importa la orientación ni posición final de Karel.
* Nunca va a haber charcos que usen la casilla inferior izquierda.
* Siempre habrá al menos un charco

**Ejemplo**

**Entrada**



**Salida**



Electricista

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Historia**

En su nuevo trabajo como electricista, Karel tiene la importante tarea de cambiar todos los focos que hay en la oficina. La oficina donde trabaja Karel es un rectángulo dibujado con paredes (puedes ver el ejemplo de abajo para entender mejor). Para los trabajadores es importante poder caminar libremente por la oficina, es por eso que, además de las paredes que dibujan el rectángulo, no existirá ninguna otra pared que te estorbe para recorrer la oficina como tu quieras.

Cada casilla dentro de la oficina tiene un foco. Si una casilla del rectángulo tiene **un zumbador** entonces el foco de esa casilla se encuentra **encendido**. Por otro lado, si la casilla no tiene **ningún zumbador** entonces el foco se encuentra **apagado**. Por cada foco en la oficina, nuestro amigo electricista tiene la tarea de: si el foco está encendido, apagarlo; si el foco está apagado, encenderlo.

Karel sabe entonces que debe recorrer toda la oficina y cambiar los focos en cada una de las casillas, si hace bien su trabajo, al final **todas las casillas deben quedar al revés de como estaban al inicio**.

**Problema**

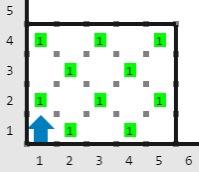
Escribe un programa que ayude a Karel a recorrer todas las casillas de la oficina y utilizar los zumbadores en su mochila para encender los focos apagados colocando un zumbador y apagar los focos encendidos recogiendo el zumbador de las casillas necesarias.

**Consideraciones**

* La oficina de Karel siempre es un rectángulo.
* Todas las casillas en la oficina tienen **0 o 1 zumbador**.
* No habrá paredes dentro de la oficina que estorben a Karel.
* Karel inicia en la posición (1, 1) del mundo orientado al norte.
* Al inicio, Karel tiene infinitos zumbadores en la mochila.
* No importa la posición ni la orientación final de Karel.

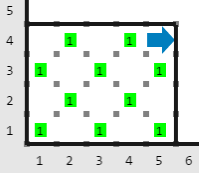
**Ejemplos**

**Entrada**



La oficina es un rectángulo de 4 filas cada una con 5 columnas. Hay 10 focos encendidos (representados con montones de 1 zumbador) y 10 focos apagados (montones de 0 zumbadores).

**Salida**



Después de que Karel apagó los focos encendidos y encendió los focos apagados la oficina se ve así.

Esquiar

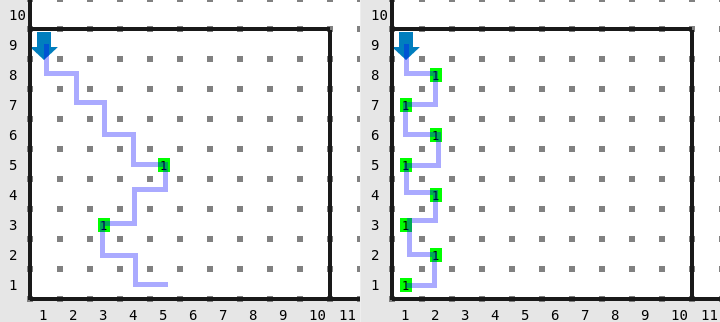
|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Historia**

Llegó el invierno y ¡a Karel le regalaron unos esquís! La montaña que está cerca de su casa está cubierta de nieve así que decide aprender a esquiar para estrenarlos.

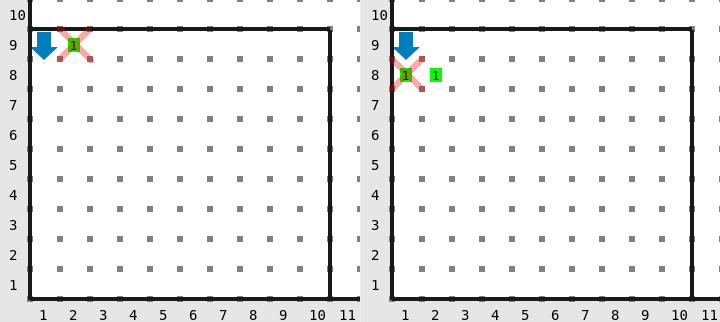
Las reglas para esquiar en esa montaña es que siempre debes bajar en diagonal. Un paso diagonal es un paso hacia abajo seguido de un paso ya sea a la izquierda o a la derecha. No toda la montaña es segura para esquiar, así que colocaron zumbadores que marcan la ruta que debe seguir Karel al bajar por la montaña. Cada vez que Karel encuentre un zumbador, debe cambiar de dirección.

Aquí hay algunos ejemplos de rutas, con la ruta que debes tomar trazada con una línea azul:



Conforme Karel baja por la montaña, quiere ayudar a otras personas que quieren aprender a esquiar marcando un poco mejor la ruta que deben seguir. Para hacer esto, deja un montón de dos zumbadores en cada casilla por la que pasa. Esto quiere decir que si la casilla ya tenía un zumbador, sólo debe dejar un zumbador más. Karel debe detenerse antes de chocar contra una pared.

No van a haber zumbadores mas que en las casillas por las que pasa Karel, ni zumbadores a la mitad de un paso en diagonal, así que estas rutas estarían mal:



La primer ruta está mal porque Karel no pasaría por ahí. La segunda ruta está mal porque no puede haber un zumbador a la mitad del paso en diagonal.

**Problema**

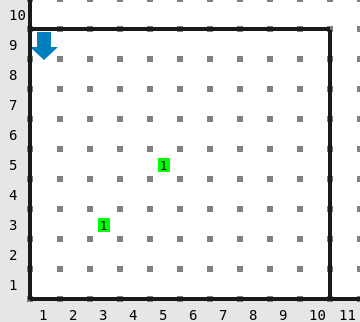
Escribe un programa que mueva a Karel desde su posición inicial, baje por la montaña usando solo pasos en diagonal y marque cada casilla que visitó con un montón de dos zumbadores.

**Consideraciones**

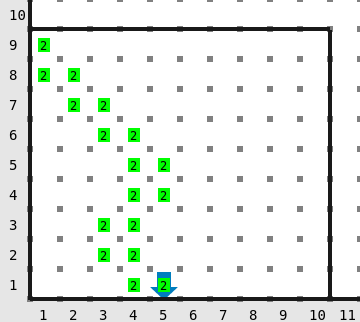
* Karel siempre inicia en la esquina de arriba a la izquierda viendo hacia el sur.
* Karel tiene infinitos zumbadores en la mochila.
* Karel debe detenerse antes de chocar contra una pared. Si choca, no va a obtener puntos.
* La casilla donde inicia Karel está vacía.
* No importa la orientación ni posición final de Karel.
* No pueden quedar zumbadores en el mundo además de los que marcan el camino.
* No habrán zumbadores que te hagan cambiar de dirección a la mitad de un paso en diagonal.

**Ejemplos**

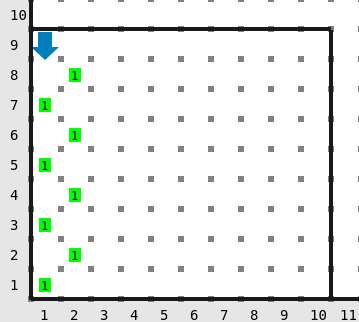
**Entrada**



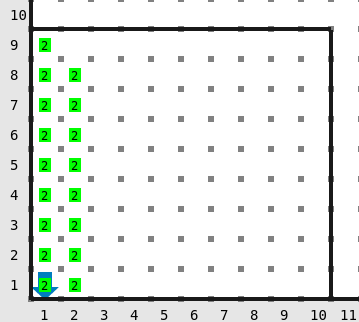
**Salida**



**Entrada**



**Salida**



Montaña

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

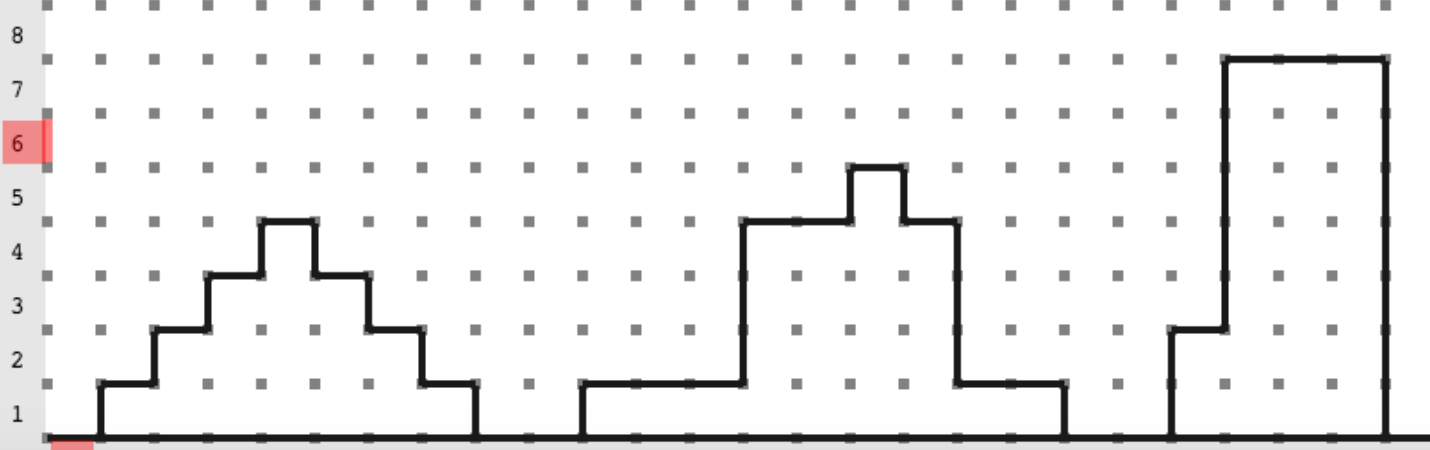
**Historia**

Karel quiere ir a una aventura. No se siente muy confiado en sus habilidades, entonces no va a matar un monstruo, o navegar el mar, porque eso sería muy complicado. En lugar de eso, Karel va a escalar una montaña. No varias montañas, porque eso le da miedito, y Karel solo está preparado para escalar una.

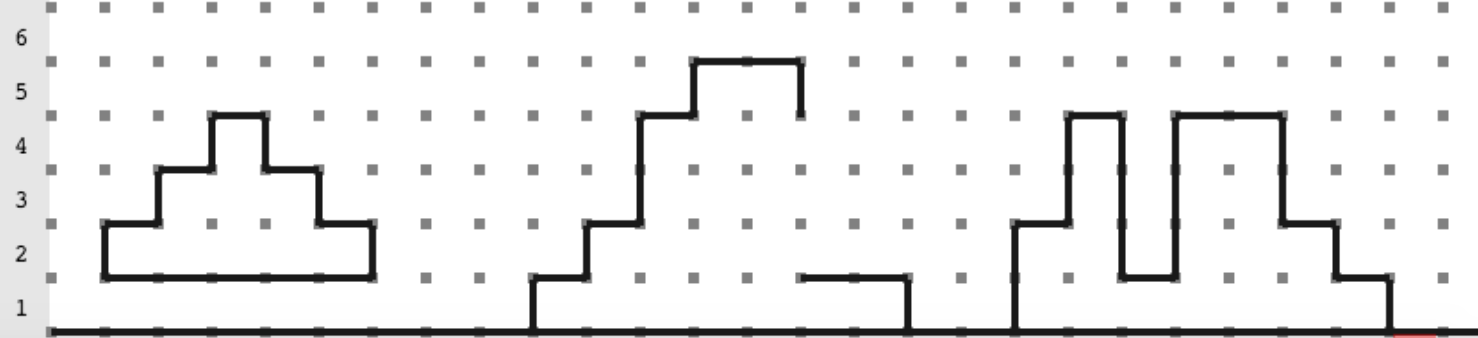
¿Qué es una montaña? Seguramente te estarás preguntando. O no. Te vamos a decir de todos modos. Una montaña tiene las siguientes características:

* Está pegada a la base del mundo de Karel (La fila 1), porque si no sería una montaña voladora.
* Es completamente cerrada, porque si no sería una cueva.
* El lado izquierdo de la montaña sube hasta llegar a la cima, y el lado derecho de la montaña baja hasta llegar a la base. Solamente existe un "pico" en la montaña, porque si no sería más de una montaña.

Estas cosas son montañas en Karel:

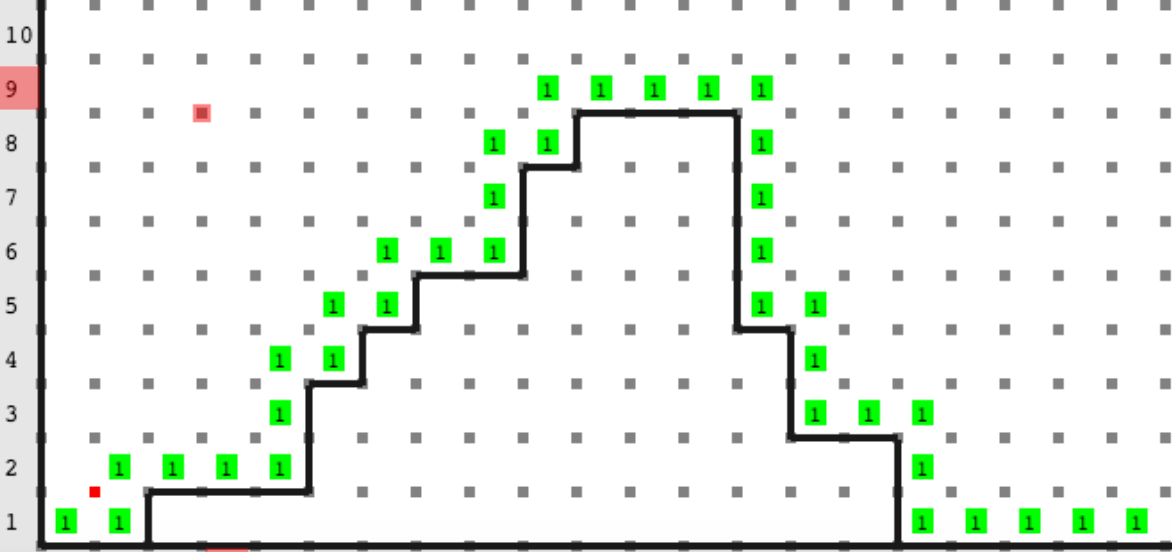


Estas cosas no son montañas en Karel:



Karel está extra emocionado por empezar su aventura porque el mundo acaba de llover zumbadores, dejando a las montañas rodeadas por montones de un beeper. (Toma en cuenta que la lluvia del mundo de Karel es rara, entonces los lados de la montaña también son rodeados por zumbadores)

Una montaña con lluvia se ve así:



Como prueba de que escaló la montaña, Karel quiere que la posición (1,1) tenga la misma cantidad de zumbadores que la altura de la montaña. Esto finalmente hará que Karel sea feliz.

**Problema**

Escribe un programa que deje la altura de la montaña representada por zumbadores en la posición (1,1).

**Consideraciones**

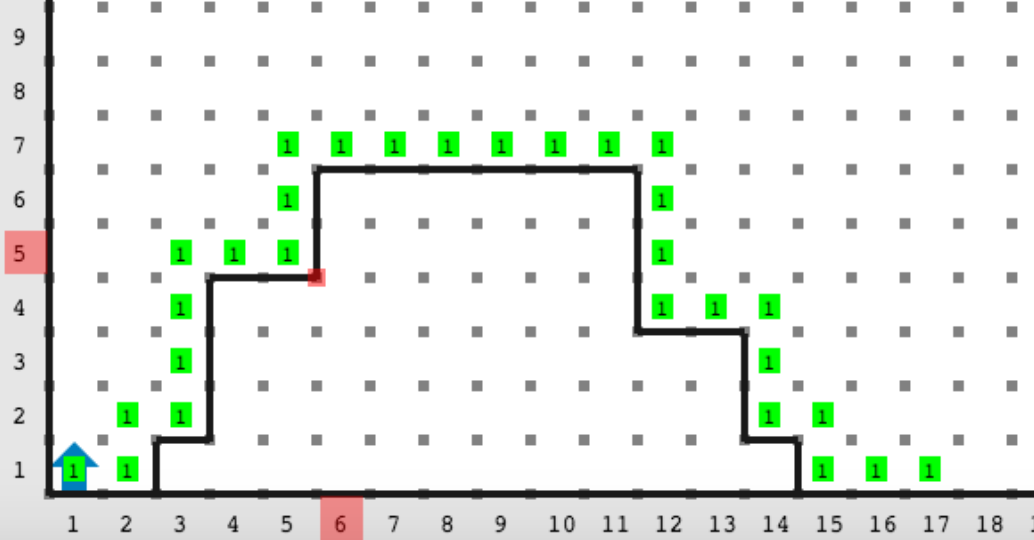
* Karel siempre inicia en la posición (1,1) viendo hacia el Norte.
* Todos los montones de zumbadores que están en la montaña son de tamaño 1 y no hay zumbadores que no sean los de la montaña.
* No importan la orientación ni la posición final de Karel.
* Solamente importan los zumbadores que están en la posición (1,1), los demás se pueden recoger o no.
* No importa la cantidad final de zumbadores en la mochila de Karel.
* La montaña siempre tendrá altura menor a 100.
* Karel no necesariamente tiene que escalar toda la montaña, solo importa que sepa la altura de la montaña.
* No hay montañas planas (De altura 0).
* La altura de una montaña es igual a la cantidad de veces que Karel tiene que moverse al norte para llegar a la cima.

**Subtareas**

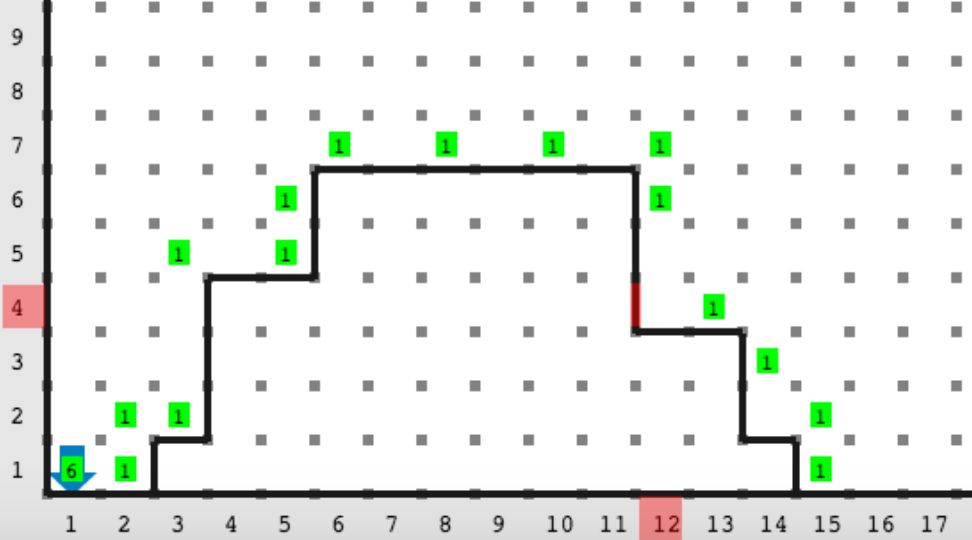
* **Subtarea 1 [26 puntos]:** Karel no tiene zumbadores en su mochila.
* **Subtarea 2 [74 puntos]:** Karel tiene zumbadores infinitos en su mochila.

**Ejemplo**

**Entrada**



**Salida**



Recuerda tu Orientación

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

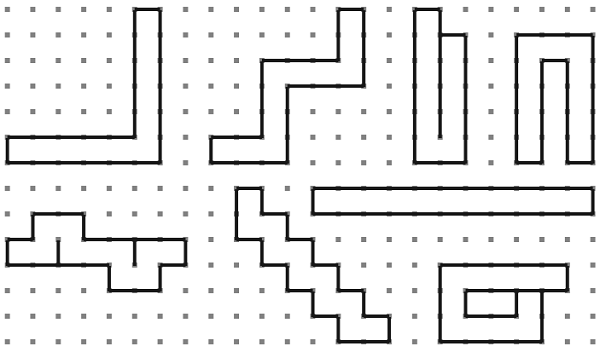
**Historia**

Los amigos de Karel han desafiado a nuestro amigo a encontrar el zumbador perdido en el pasillo donde vive. Karel está cansado de escuchar que es un robot muy desorientado, por ello quiere darles una lección.

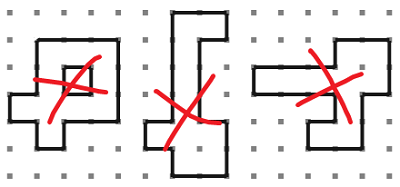
**¿No sabes lo que es un pasillo? No hay problema, te explicamos.** Imagina una línea de varias casillas del mundo dibujada con paredes como se muestra en el dibujo de abajo:

Ejemplo de una línea

Puedes doblar la línea anterior cuantas veces quieras para formar diversos pasillos. Recuerda que la línea de casillas sin doblar también cuenta como un pasillo. Puedes dibujar pasillos de muchas figuras diferentes, a continuación te mostramos solo algunos ejemplos:



Sin embargo, **no puedes doblar la línea de tal forma que se cruce consigo misma**. En seguida te mostramos algunos ejemplos de líneas que se cruzan a si mismas y por lo tanto **NO son pasillos:**



Karel inicia en uno de los extremos del pasillo; **puedes reconocer un extremo del pasillo por ser una casilla rodeada de 3 o 4 paredes**. Dentro de las casillas del pasillo existe un zumbador perdido. Para mostrarle a sus amigos que no es un robot desorientado, Karel quiere pararse sobre el zumbador perdido y mirar en la misma dirección a la que Karel estaba viendo en la casilla donde comenzó.

Por ejemplo, imagina que Karel se encuentra en un extremo del pasillo mirando al norte. Karel quiere caminar por el pasillo hasta encontrar el zumbador y ahí detenerse. Una vez que Karel esté parado sobre el zumbador, él quiere mirar al norte, justo hacia donde estaba mirando al iniciar su búsqueda.

**Problema**

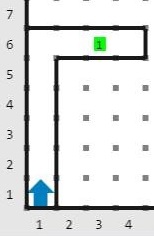
Escribe un programa que mueva a Karel desde su posición inicial en el pasillo hasta el zumbador perdido. **Recuerda que Karel debe terminar su ejecución orientado en la misma dirección con la que inició.**

**Consideraciones**

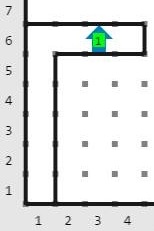
* Karel siempre inicia en un extremo del pasillo.
* Al inicio, Karel no tiene zumbadores en la mochila.
* Siempre existirá **un solo zumbador dentro del pasillo**.
* Karel inicia mirando a la siguiente casilla por la que debe caminar.
* Puede ser que el pasillo donde vive Karel sea una sola casilla.
* **Solo importa la posición y orientación final de Karel**.

**Ejemplo**

**Entrada**



**Salida**



En el ejemplo anterior, Karel inicia mirando al norte. Después de caminar derecho, girar a la derecha y seguir caminando hasta encontrar el zumbador perdido, Karel se detiene y voltea de nuevo al norte para terminar su ejecución orientado en la misma dirección que tenía al iniciar su búsqueda.