Fix it Karel

Historia

El maléfico Chuzpa, eterno enemigo de Karel, se ha dado últimamente a la tarea de destruir monumentos. Karel, por su parte, se dedica a restaurar y reconstruir todo lo que Chuzpa destruye.

Chuzpa destruyó una antigua pirámide de Karelotitlán. Originalmente la pirámide estaba erigida sobre un rectángulo de ***m*** x ***n***. Utilizando su rayo vaporizador, Chuzpa vaporizó por completo una columna y una fila de la pirámide con lo cual ésta quedó en un rectángulo de ***(m-1)*** x ***(n-1)***. Al hacerlo, la pirámide se desplomó. Los acomedidos ciudadanos de Karelotitlán rápidamente recogieron los escombros.

Al llegar Karel a reconstruir la pirámide, nadie supo decirle cuál era el tamaño original de la misma. Nadie recordaba el valor de ***m*** ni de ***n***. Sin embargo recordaban el área que limpiaron, es decir, el área que ocupaba el rectángulo de ***(m-1)*** x ***(n-1)***.

Para evaluar el material que necesitará, Karel quiere saber, en base al área que ocupa el rectángulo de ***(m-1)*** x ***(n-1)***, ¿Cuál es el área mínima y máxima que pudo haber ocupado la pirámide original?

Problema

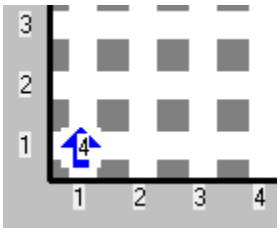
Escribe un programa que, sabiendo Karel el área que ocupaba el rectángulo de ***(m-1)*** x ***(n-1)***, determine cuál es el área mínima y máxima que pudo haber ocupado el rectángulo de ***m*** x ***n***.

Consideraciones

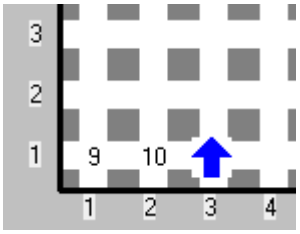
- Karel empieza en la esquina inferior izquierda viendo al norte.
- La casilla (1,1) contiene un número de zumbadores igual al área que ocupaba el rectángulo de ***(m-1)*** x ***(n-1)***.
- El mundo es de 100 x 100, no tiene paredes internas ni zumbadores aparte de los de la casilla (1,1)
- Karel lleva infinitos zumbadores en la mochila.
- El área del rectángulo ***(m-1)*** x ***(n-1)*** es menor a 100 y mayor que 0.
- Tu programa deberá dejar en la casilla (1,1) un número de zumbadores igual al área mínima que pudo haber ocupado el rectángulo de ***m*** x ***n***.
- Tu programa deberá dejar en la casilla (2,1) un número de zumbadores igual al área máxima que pudo haber ocupado el rectángulo de ***m*** x ***n***.
- No importan la posición ni la orientación final de Karel, tampoco los zumbadores en casillas distintas a las mencionadas (1,1) y (2,1).

Ejemplo

Entrada



Salida



El rectángulo de ***(m-1)*** x ***(n-1)*** ocupaba un área de **4**. Eso implica que el área mínima que pudo haber ocupado el rectángulo original es **9** (en este caso el rectángulo original sería de **3 x 3**) y el área máxima que pudo haber ocupado es **10** (en este caso el rectángulo original sería de **5 x 2**).

Karel Attractio

Historia

Después de jugar Attractio (http://www.gamecoderstudios.com/Attractio), Karel quedó muy emocionado y decidió hacer sus propios experimentos de alteración de gravedad.

Karel ha creado un mundo en donde hay montones que atraen a Karel con una fuerza gravitacional igual al número de zumbadores en el montón. Karel puso estos montones a todo lo largo de la fila 1 del mundo, pero no contaba con que una vez colocados, la fuerza gravitacional le haría muy difícil moverse.

Si Karel tiene montones a su derecha y a su izquierda, la fuerza gravitacional de unos y otros se anula. De modo que el único lugar en que Karel está libre de fuerza gravitacional es donde la suma de las fuerzas de los montones a su izquierda es igual a la suma de las fuerzas de los montones a su derecha. **Si Karel está sobre un montón, se considera que este montón lo está atrayendo hacia la izquierda.**

Problema

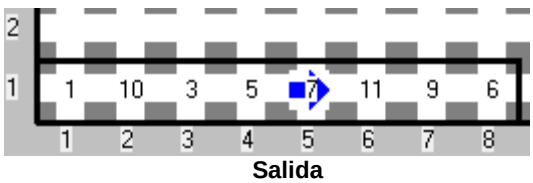
Escribe un programa que, colocados los montones de zumbadores, posicione y apague a Karel en la casilla de la fila 1 donde las fuerzas gravitacionales sobre él se anulan (ver ejemplo).

Consideraciones

- Karel inicia en la posición (1,1) del mundo viendo al Norte.
- La fila 1 del mundo está llena de montones de zumbadores, ningún montón tiene más de 100 ni menos de 1.
- Karel inicia con **0 zumbadores en la mochila**.
- Los mundos tienen sólo **1 fila de alto**.
- Para obtener los puntos en un caso, tu programa deberá apagar a Karel en la posición donde las fuerzas gravitacionales sobre él se anulen (**se asegura que siempre habrá una casilla dónde eso suceda**), *no importan la dirección final de Karel ni los montones de zumbadores que queden en el mundo.*
- ***IMPORTANTE***
  - En casos que valen el **32%** de los puntos, Karel podrá ejecutar la instrucción ***avanza/move*** como máximo 500 veces. Si en algún caso tu programa ejecuta la instrucción avanza/move más de 500 veces obtendrás ***0* puntos para ese caso**.
  - En los casos que valen el restante **68%** de los puntos, Karel no tiene ese límite.

Ejemplo

Entrada



Salida

Recuerda que si Karel está sobre un montón se considera que el montón lo está atrayendo hacia la izquierda. Para este ejemplo Karel se detiene en la casilla (5,1), eso quiere decir que la fuerza que lo atrae hacia la **izquierda** es igual a: **1+10+3+5+7=26**. La fuerza que lo atrae hacia la **derecha** es igual a: **11+9+6=26**. De modo que las fuerzas a la izquierda y derecha se anulan mutuamente y Karel puede descansar tranquilamente.

Capitán Karel

Historia

El Capitán Karel tiene un nuevo escudo de *limodium* para vencer a sus enemigos. El *limodium* es muy resistente pero muy pesado, para lanzarlo el Capitán Karel requiere de un dispositivo de propulsión auxiliar que se calibra para que el escudo avance exactamente ***K*** casillas.

El Capitán Karel tiene el mapa de su siguiente misión. En el mapa se muestra a los enemigos como montones de un zumbador. La energía del dispositivo de propulsión alcanza para lanzar su escudo ***1 vez por columna***. El Capitán puede elegir desde qué fila lanzarlo y con qué dirección (arriba o abajo). Cuando el Capitán lanza su escudo elimina a todos los enemigos en esa columna que estén en la casilla desde dónde se lanzó el escudo hasta una distancia ***K*** en la dirección en que se lanzó (ver ejemplo).

Para optimizar el consumo de energía debes ayudar al Capitán Karel a calcular la ***K*** mínima necesaria para poder eliminar a ***todos*** los enemigos que aparecen en el mapa.

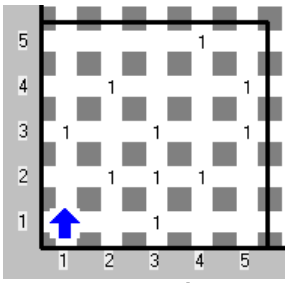
Problema

Escribe un programa que ayude a Karel a calcular la ***K*** mínima necesaria para eliminar a todos los enemigos del mapa. Una vez calculada, tu programa debe apagar a Karel en la casilla (***K***,1).

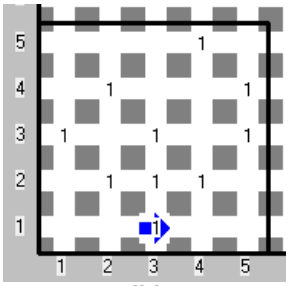
Consideraciones

- Karel puede iniciar en cualquier lugar del mundo y con cualquier orientación.
- El valor mínimo que se puede calibrar en el escudo para ***K*** es ***1***.
- El mundo es rectangular, sin paredes internas. El número de filas es menor o igual que el de columnas.
- Las columnas pueden tener cualquier número de enemigos, desde 0 hasta el alto del mundo.
- Para obtener los puntos, tu programa deberá dejar a Karel en la casilla (***K***,1). *No importan la orientación final de Karel ni los zumbadores que queden en el mundo.*
- ***IMPORTANTE***
  - En un conjunto de casos que valen el **27%** de los puntos, habrá siempre **2 enemigos por columna** y Karel tendrá **infinitos zumbadores en la mochila.**
  - En otro conjunto que vale **30%** de los puntos, puede haber **cualquier cantidad de enemigos por columna** y Karel tendrá **infinitos zumbadores en la mochila.**
  - En el último conjunto con valor de **43%** de los puntos, puede haber **cualquier cantidad de enemigos por columna** y Karel inicia con **0 zumbadores en la mochila.**

Ejemplo



Entrada



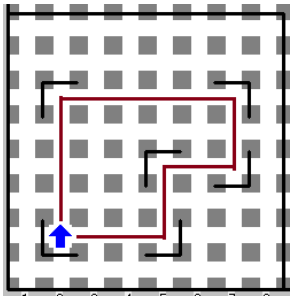
Salida

Para la columna 4 se requiere una ***K*** de 3 (*si Karel lanza el escudo desde la posición (4,2) con dirección arriba*), para cualquier otra columna basta con una ***K*** menor a 3. Por lo tanto la ***K*** mínima necesaria es de ***3*** y Karel debe terminar en la casilla **(3,1)**

Mr. Karel-tastic

Historia

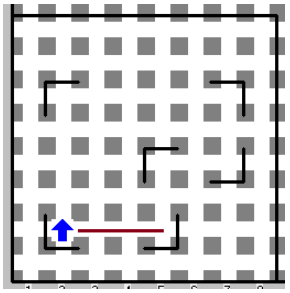
Karel tiene el súper poder de estirar su cuerpo. Su poder es limitado y puede estirarse a lo más ***K*** casillas. Además, Karel está diseñando un nuevo cuarto de entrenamiento. El cuarto tiene internamente varias paredes en forma de **L** (ver figura). El cuarto se usa así:



1. Karel inicia a correr en una de las **L**'s viendo al norte.
2. Cada que Karel tope con una **L** deberá girar hacia la izquierda o hacia la derecha, dependiendo de que lado tenga libre. *Siempre tendrá un lado libre*.
3. Karel debe seguir corriendo en la nueva dirección hasta topar con la siguiente **L,** girar y continuar así indefinidamente. En la figura puede verse un cuarto de ejemplo y el circuito que debe correr Karel.

El recorrido del cuarto de entrenamiento siempre es un circuito, es decir, seguir las instrucciones de arriba siempre te lleva de nuevo al punto donde iniciaste, además se asegura que Karel sólo topa con **L**'s. Sin embargo a veces el camino puede cruzarse.

Karel desea practicar mientras usa su súper poder, es decir, mientras su cuerpo tiene un largo ***K***, pero le preocupa que en algún momento, mientras está *estirado*, pueda chocar contra el mismo (ver ejemplos).



Para saber el largo ***K*** al que puede estirarse Karel debes medir la distancia entre su casilla de inicio y la primera **L** a su derecha (Para que sea más claro, observa la figura). En la figura puede observase que hay 4 casillas entre la posición donde inicia Karel y la primera **L** a su derecha, por lo tanto, para este ejemplo, ***K=4***.

Debes ayudar a Karel a saber si es seguro para él recorrer el circuito en el cuarto de entrenamiento mientras su cuerpo está estirado.

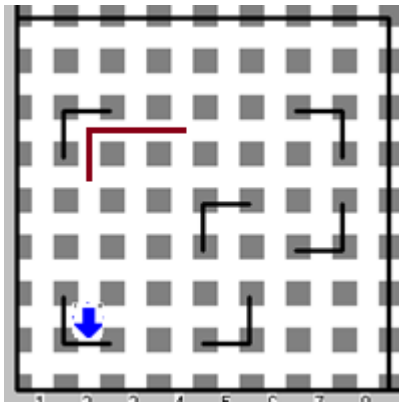
Problema

Escribe un programa que determine si Karel puede recorrer el circuito con su cuerpo estirado sin chocar contra él mismo. Tu programa deberá dejar a Karel viendo al **sur si es posible** y viendo al **norte si choca con su propio cuerpo.**

Consideraciones

- Karel inicia en una **L** viendo al norte y con el frente y la derecha libre.
- El cuarto de entrenamiento nunca es mayor a **15** x **15**.
- Para este programa en el entorno de evaluación Karel podrá ejecutar más de 10,000,000 de operaciones.
- Para obtener los puntos, tu programa deberá dejar a Karel viendo al **sur** si es seguro correr el circuito con su cuerpo estirado y al **norte si corriendo el circuito estirado choca con él mismo.** *No importan la posición final de Karel ni los zumbadores que queden en el mundo.*
- ***IMPORTANTE***
  - En un conjunto de casos que valen el **53%** de los puntos, Karel inicia con **infinitos zumbadores en la mochila.**
  - En otro conjunto que vale **29%** de los puntos, Karel inicia con **1 zumbador en la mochila.**
  - En el último conjunto con valor de **18%** de los puntos, Karel inicia con **0 zumbadores en la mochila.**
  - En cada uno de los conjuntos de casos es necesario resolver correctamente **todos** los casos para obtener puntos, es decir, todos los casos del conjunto están agrupados.

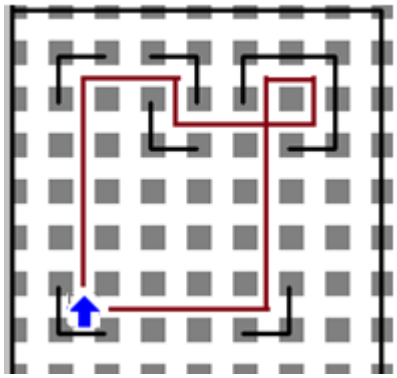
Ejemplos



En este ejemplo ***K=4*** porque hay 4 casillas entre la posición donde inicia Karel y la primera **L** que hay a su derecha. En la figura se muestra a Karel con su cuerpo estirado 4 casillas recorriendo el circuito. Para visualizarlo mejor, puedes imaginar que Karel es una *serpiente* de largo ***K*** que va avanzando por el circuito.

Este circuito es seguro para ser corrido por Karel ya que el circuito nunca se cruza sobre sí mismo, de modo que Karel nunca chocará contra su propio cuerpo

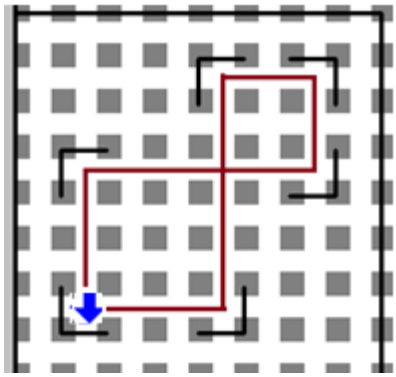
Karel debe terminar viendo al SUR, como se muestra.



En este ejemplo ***K=5*** porque hay 5 casillas entre la posición donde inicia Karel y la primera **L** que hay a su derecha.

Este circuito tiene un cruce, si observas el cruce verás que con su cuerpo en largo 5 (o mayor), Karel chocaría contra si mismo, de modo que este circuito **no es seguro. (Para este ejemplo, si *K* fuera menor o igual que 4, no chocaría contra si mismo)**

Karel debe terminar viendo al NORTE, como se muestra.



En este ejemplo ***K=4*** porque hay 4 casillas entre la posición donde inicia Karel y la primera **L** que hay a su derecha.

Este circuito tiene un cruce, sin embargo el largo mínimo que se requeriría para que Karel chocara contra él mismo es de 9, por lo tanto este circuito es seguro para Karel.

Karel debe terminar viendo al SUR, como se muestra.