

## Flappy Karel

### Historia

Regresando de un retiro espiritual donde estuvo alejado de la tecnología, Karel se enteró que era demasiado tarde para bajar el juego de *Flappy Bird*. Triste, pero no derrotado Karel decidió que no iba a quedarse sin el último juego de moda y por lo tanto va a programar su propio *Flappy Karel*.

El objetivo de *Flappy Karel* es que Karel, partiendo del extremo izquierdo del mundo, avance hacia la derecha lo más posible sin chocar con ningún obstáculo. Antes de avanzar una columna hacia la derecha, Karel puede realizar una de dos acciones, **subir EXACTAMENTE 2 filas** o **bajar EXACTAMENTE 1 fila**. Después de subir o bajar, Karel avanza una columna a la derecha.

En el mundo hay obstáculos, un obstáculo es una pared vertical que va del extremo superior al extremo inferior del mundo y que en algún lugar tiene una abertura con ancho mayor o igual a una fila. Para avanzar, Karel debe tomar las decisiones correctas para poder pasar a través de esa abertura.

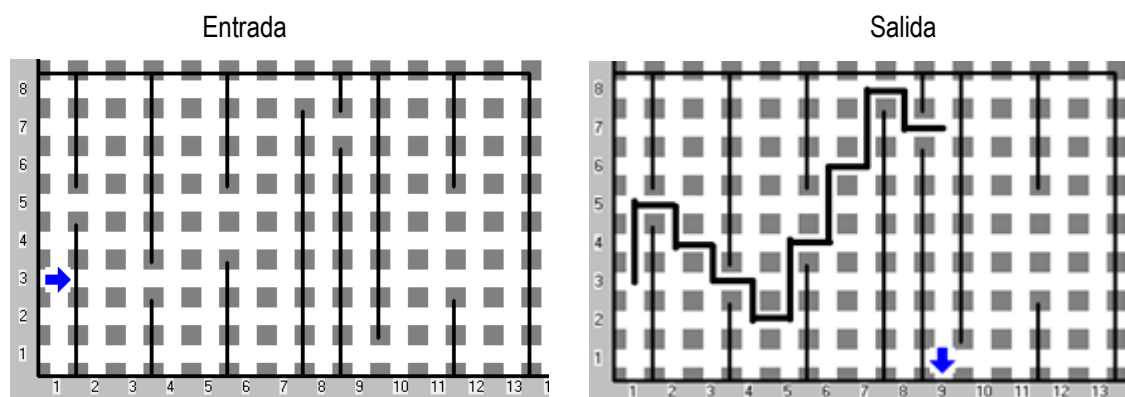
### Problema

Escribe un programa que, dado un mundo con obstáculos decida en cada columna si Karel debe subir o bajar de modo que avance lo más posible hacia la derecha, cuando llegue a una columna donde ya no sea posible avanzar, Karel deberá bajar hasta la fila 1 y ahí apagarse.

### Consideraciones

- Karel empieza en algún lugar de la columna más a la izquierda del mundo orientado al este.
- Karel inicia con infinitos zumbadores en la mochila.
- El mundo es rectangular y no tiene ningún zumbador.
- Para obtener los puntos en un caso **Karel debe apagarse en la fila 1 de la columna más a la derecha a la que pueda llegar (La fila 1 es la fila inferior del mundo).**
- No importan la orientación final de Karel ni los zumbadores que dejes en el mundo.
- **IMPORTANTE:** Para evitar que tu programa trate de *adivinar* la columna el comité evaluador agrupará casos de prueba, para obtener los puntos de un grupo de casos tu programa deberá resolver correctamente **todos** los casos de prueba del grupo.

### Ejemplo



El ejemplo de entrada muestra un mundo de Karel con obstáculos. El mundo de salida muestra un posible camino que pudo seguir Karel, el camino es el siguiente: **en la columna 1, sube; en la columna 2 baja, en la 3 baja, 4 baja, 5 sube, 6 sube, 7 sube, 8 baja. En la columna 9 ya no es posible seguir avanzando ya que si subes chocas con la pared superior y si bajas no alcanzas a llegar a la abertura por lo que debe bajar a la fila 1 de esa columna y apagarse.**

**NOTA:** Para avanzar a la derecha es posible que existan varios caminos, lo único que importa para tu puntaje es que Karel se apague en la columna correcta, no importa el camino como llegó ahí.

## Karel el demócrata

### Historia

Karel sufre de una fiebre democrática, ahora le gusta organizar votaciones para todo. Pero Karel es muy estricto y quiere que en sus votaciones el ganador siempre tenga *estricta mayoría*, es decir, alguien es ganador sólo si recibe una cantidad de votos mayor que la mitad del número de votantes.

Karel necesita implementar un sistema de conteo de votos el cual reciba como entrada los votos y decida si hay un ganador que tenga *estricta mayoría*. Para este sistema Karel recibirá un mundo de tamaño  $N$  columnas de ancho por **100** filas de alto. En la fila inferior habrá montones de zumbadores en todas las columnas, estos montones pueden tener entre 1 y 99 zumbadores. Cada montón representa un voto, el número de zumbadores en el montón representa la opción elegida por el votante.

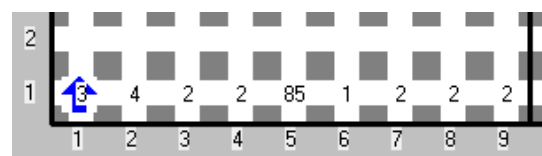
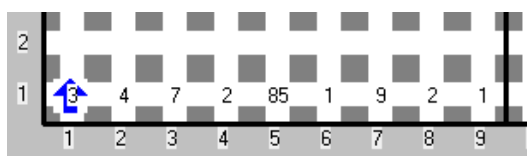
### Problema

Escribe un programa que reciba un mundo con votos y decida si hay una opción que sea ganadora con *estricta mayoría*. Si hay una opción ganadora, Karel deberá terminar con tantos zumbadores en la mochila como el número de esa opción, es decir, *si la opción 5 es ganadora, Karel deberá terminar con 5 zumbadores en la mochila*. Si no hay opción ganadora Karel deberá terminar sin zumbadores en la mochila.

### Consideraciones

- Karel inicia en la posición (1,1) del mundo viendo al Norte.
- La fila 1 del mundo está llena de montones de zumbadores, los montones tienen entre 1 y 99 zumbadores.
- Karel inicia con **0 zumbadores en la mochila**.
- Los mundos tienen siempre **100 filas de alto**.
- No importan la dirección ni la orientación final de Karel, tampoco los zumbadores que queden en el mundo
- Para cada caso, Karel debe terminar con **0 zumbadores en la mochila si no hay opción ganadora** y con  **$k$  zumbadores en la mochila, si  $k$  es la opción ganadora**.
- En este problema se agruparan casos de prueba, para obtener los puntos de un grupo, tu programa deberá resolver de manera correcta **todos los casos del grupo**.
- **IMPORTANTE**
  - Para un grupo de casos con valor de 33 puntos los votos estarán ordenados
  - Para otro grupo de casos con valor de 10 puntos los votos estarán desordenados
  - Para un grupo más con valor de 20 puntos tu programa deberá utilizar la instrucción **avanza/move** un máximo de  **$4N$**  veces (donde  $N$  es el número de columnas del mundo).
  - Para un último grupo con valor de 37 puntos tu programa deberá utilizar la instrucción **avanza/move** un máximo de  **$2N$**  veces (donde  $N$  es el número de columnas del mundo).

### Ejemplos



En los ejemplos se muestran 2 mundos de entrada. El primer mundo no tiene ninguna opción ganadora, por lo que Karel deberá terminar con **0 zumbadores en la mochila**. En el segundo mundo **la opción 2 es ganadora**, por lo que Karel deberá terminar con **2 zumbadores en la mochila**.

## Zig-zag de estaturas

### Historia

En primaria Karel gustaba de hacer desesperar a sus maestras. Cuando éstas ordenaban a los niños que se formaran por estaturas, Karel organizaba a sus compañeros para que hicieran una *fila de estaturas en zig-zag*, esto es, los estudiantes en una posición non de la fila debían ser más bajos que los estudiantes en la posición delante de ellos (a excepción del estudiante en la posición 1 de la fila que no tenía a nadie delante). Y los estudiantes en posiciones pares en la fila debían ser más altos que el estudiante en la posición delante de ellos. Es decir la fila debe ir *bajo, alto, bajo, alto, bajo, alto,...*

Para este problema Karel recibirá un mundo de tamaño  $N$  columnas de ancho por **100** filas de alto. En la fila inferior habrá montones de zumbadores en todas las columnas, estos montones pueden tener entre 1 y 99 zumbadores. Cada montón representa un niño, el número de zumbadores en el montón representa la altura de ese niño.

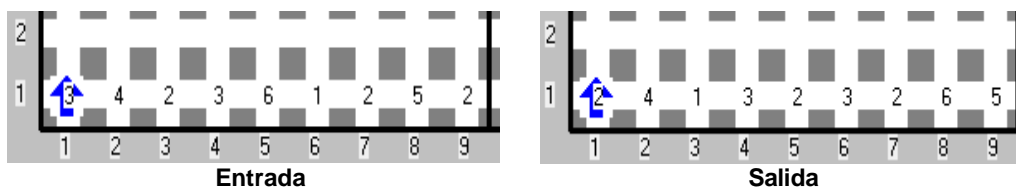
### Problema

Escribe un programa que dados los niños representados por los montones de zumbadores de la fila 1, los reordene de modo que formen una *fila de estaturas en zig-zag*.

### Consideraciones

- Karel inicia en la posición (1,1) del mundo viendo al Norte.
- La fila 1 del mundo está llena de montones de zumbadores, los montones tienen entre 1 y 99 zumbadores.
- Karel inicia con **0 zumbadores en la mochila**.
- Los mundos tienen siempre **100 filas de alto**.
- No importan la dirección ni la orientación final de Karel.
- **PARA UN MUNDO PUEDE HABER MUCHAS FILAS POSIBLES QUE SON CORRECTAS, CUALQUIERA DE ELLAS QUE DEJE TU PROGRAMA SERÁ CONSIDERADA CORRECTA.**
- **TE ASEGURAMOS QUE SIEMPRE HABRÁ AL MENOS UNA SOLUCIÓN POSIBLE.**

### Ejemplo



El mundo de ejemplo muestra un grupo inicial de niños, sus alturas representadas por el número de zumbadores en cada montón. El mundo de salida muestra como quedaría una posible fila por estaturas en zig-zag. Además de la solución que se muestra en el mundo de salida hay muchas otras posibles.

## Karel Gump

### Historia

Karel Gump, un primo Kareleano de Forrest Gump heredó también el gusto por correr. Como siempre, el mundo de Karel es distinto al mundo real. Karel Gump debe seguir las siguientes reglas al correr:

- Avanzar hasta encontrar una casilla que tenga zumbador, al llegar a esa casilla Karel debe girar a la izquierda.
- Si Karel choca con una pared debe detenerse.
- Si Karel pasa por una casilla por la que ya había pasado antes, debe detenerse.

El mundo donde corre Karel es rectangular y no tiene paredes internas, sin embargo, hay casillas que tienen 1 zumbador y que hacen que al encontrarlas Karel tenga que girar. **Karel desea maximizar el largo de su recorrido antes de tener que detenerse**, para lograr lo anterior Karel planea agregar algunos zumbadores al mundo de modo que al hacer su carrera, el largo del recorrido que hace antes de detenerse sea el mayor posible.

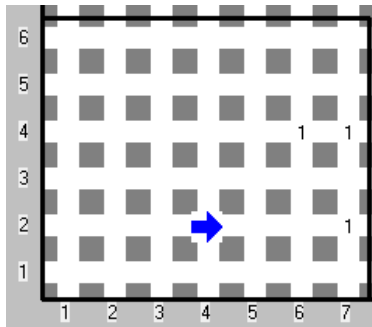
### Problema

Escribe un programa que dado el mundo, las casillas con zumbador iniciales y la posición y orientación inicial de Karel, **coloque zumbadores en casillas, sin alterar los zumbadores que ya están, de modo que el recorrido de Karel sea el mayor posible.**

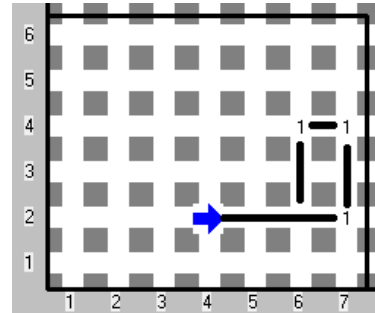
### Consideraciones

- Karel inicia en una casilla del mundo que **no tiene zumbadores** con alguna orientación.
- Karel inicia con un número infinito de zumbadores en la mochila.
- El mundo nunca es mayor a **30 x 30**.
- El mundo no tiene paredes internas pero hay casillas que tienen **1 zumbador**.
- Para obtener los puntos, tu programa deberá dejar, sin alterar las casillas donde originalmente hay zumbador en el mundo, montones de **1 zumbador** en posiciones que mejoren el largo del recorrido que hace Karel. *No importan la posición ni la orientación final de Karel.*
- **IMPORTANTE**
  - El número de puntos que obtengas en este problema dependerá de que tan largo sea el recorrido de Karel con los zumbadores que agregaste.
  - Si el largo del recorrido es menor o igual al largo sin haber agregado ningún zumbador, obtendrás **0 puntos en ese caso**.
  - Si el largo del recorrido es igual al más largo posible, obtendrás el **100% de los puntos de ese caso**.
  - Si el largo de tu recorrido está entre los dos números anteriores, obtendrás una **parcialidad de los puntos de ese caso**.
  - Si tu programa tarda más de 30 segundos en ejecutarse obtendrás **0 puntos por ese caso**.
  - Si al final de tu programa alguna de las casillas que tenía zumbador originalmente queda sin zumbador **no obtendrás puntos por ese caso**.

## Ejemplos

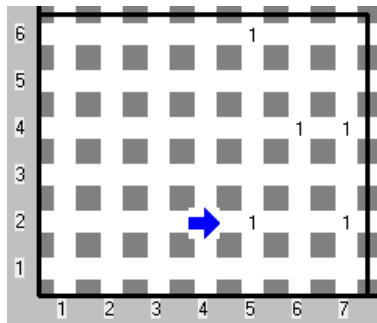


Mundo original

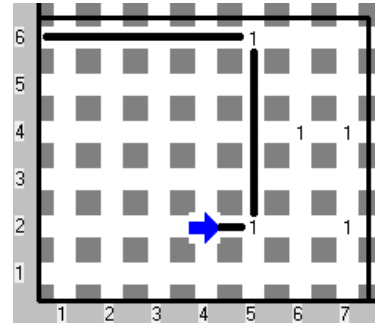


Recorrido original

Este es el mundo original donde va a correr Karel Gump, a la derecha se muestra el recorrido. Karel debe detenerse en la casilla (6,2) porque el recorrido pasa por una casilla que ya se había visitado. El recorrido tiene un largo de 8 casillas.

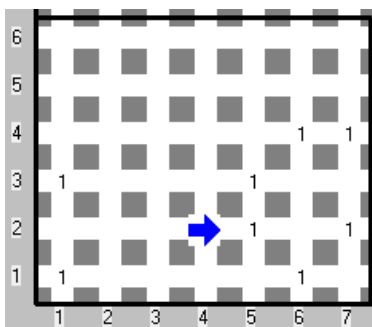


Ejemplo de modificación 1

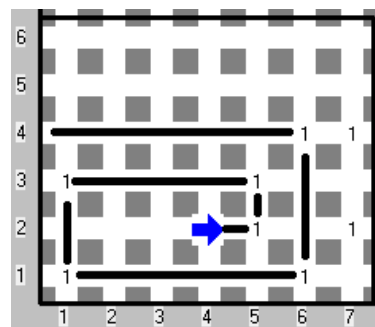


Recorrido con la modificación 1

Este ejemplo muestra una modificación que puede hacer Karel, en la izquierda muestra el mundo modificado **fíjate que no se alteraron los lugares que originalmente tenían zumbador**. A la derecha se muestra el recorrido sobre el mundo modificado, el largo del recorrido sobre el mundo modificado es de 9 casillas.



Ejemplo de modificación 2



Recorrido con la modificación 2

Este ejemplo muestra otra posible modificación mejor que la primera. En la izquierda se muestra el mundo modificado. A la derecha se muestra el recorrido sobre el mundo modificado, en este caso el recorrido tiene un largo de 21 casillas.

**La solución óptima para este mundo es un recorrido de 30 casillas.**