

Encuentra la caja

Nombre del problema	Find the Box
Límite de tiempo	1 segundo
Límite de memoria	1 gigabyte

Maj es una investigadora de robótica que trabaja en la Universidad de Lund. Maj se enteró de que existe un valioso tesoro en el sótano de la universidad. El tesoro está en una caja ubicada en un cuarto vacío que está en la profundidad. Desafortunadamente, Maj no puede simplemente ir a buscar la caja. El sótano está muy oscuro y si va con una lámpara levantaría sospechas. Su única forma de encontrar el tesoro es controlar remotamente una aspiradora robot que vive en el sótano.

El sótano está representado como una matriz de tamaño $H \times W$, en la que las filas están numeradas de 0 a $H - 1$ (de arriba hacia abajo) y las columnas están numeradas de 0 a $W - 1$ (de izquierda a derecha), lo cual significa que la celda superior izquierda es $(0, 0)$ y la celda inferior derecha es $(H - 1, W - 1)$. La caja con el tesoro está en una celda desconocida, diferente a $(0, 0)$. Todas las noches la aspiradora robot empieza en la celda superior izquierda y se mueve dentro del sótano.

Cada noche, Maj le puede dar al robot una secuencia de instrucciones de cómo debe moverse a través de una cadena formada por los caracteres " $<$ ", " $>$ ", " \wedge " y " \vee ". Formalmente, si el robot está en la celda (r, c) que está libre hacia cualquier dirección, " $<$ " mueve al robot hacia la izquierda, a la celda $(r, c - 1)$; " $>$ " mueve el robot hacia la derecha, a la celda $(r, c + 1)$; " \wedge " mueve el robot hacia arriba, a la celda $(r - 1, c)$; y " \vee " mueve al robot hacia abajo, a la celda $(r + 1, c)$.

Las paredes del sótano son sólidas, así que si el robot intenta moverse hacia afuera de la matriz, no ocurre nada. La caja también es sólida y no se puede empujar. Al final de cada noche, el robot reportará su ubicación y luego regresará a la celda superior izquierda.

El tiempo es importante, por lo que Maj decide encontrar la caja en la menor cantidad de noches posible.

Interacción

Este es un problema interactivo.

- Tu programa debe empezar leyendo una línea que contiene dos enteros H y W : la altura y el ancho de la matriz. La caja nunca estará situada en la celda de inicio del robot $(0, 0)$.
- Después tu programa debe interactuar con el evaluador. En cada ronda de la interacción, debes imprimir un signo de interrogación "?", seguido de una cadena no vacía s formada por los caracteres "<", ">", "^", "v". La longitud de esta cadena puede ser a lo más 20 000. Después tu programa debe leer dos enteros r, c ($0 \leq r \leq H - 1$, $0 \leq c \leq W - 1$), la ubicación del robot después de ejecutar las instrucciones. Ten en cuenta que el robot siempre regresa a $(0, 0)$ después de cada consulta.
- Cuando sepas la ubicación de la caja, imprime "!" seguido de los dos enteros r_b, c_b , la fila y la columna de la celda donde está la caja ($0 \leq r_b \leq H - 1$, $0 \leq c_b \leq W - 1$). Después de esto tu programa debe terminar sin hacer ninguna otra consulta. Esta última salida no cuenta como una consulta para determinar tu puntaje.

Asegúrate de hacer flush del stream de salida después de hacer una consulta, de lo contrario tu programa puede obtener el veredicto Time Limit Exceeded.

En Python, `print()` hace flush automáticamente. In C++, `cout << endl;` también hace flush además de imprimir una línea nueva; si usas `printf` deberás usar `fflush(stdout)`.

El evaluador no es adaptativo, lo que significa que la posición de la caja se determina antes de que comience la interacción.

Límites y evaluación

- $1 \leq H, W \leq 50$.
- La caja nunca va a estar ubicada en $(0, 0)$. Esto significa que $H + W \geq 3$.
- Cada consulta puede consistir de a lo más 20 000 instrucciones.
- Puedes hacer hasta 2 500 consultas.

Tu solución se evaluará con un conjunto de grupos de casos de prueba. Si tu solución falla en *cualquiera* de estos casos de prueba (por ejemplo, reporta una posición errónea de la caja (Wrong Answer), trueno (Runtime Error), excede el tiempo límite (Time Limit Exceeded), etc), recibirás 0 puntos y el veredicto correspondiente.

Si tu programa encuentra la posición de la caja exitosamente en *todos* los casos de prueba, obtendrás el veredicto Accepted y tu puntaje será calculado de la siguiente manera:

$$\text{puntaje} = \min\left(\frac{100\sqrt{2}}{\sqrt{Q}}, 100\right) \text{ puntos,}$$

donde Q es el máximo número de consultas usadas en cualquier caso de prueba. Imprimir la respuesta final no cuenta como una consulta. El puntaje se redondeará al siguiente entero.

Para recibir 100 puntos, tu programa debe resolver cada caso de prueba usando a lo más $Q = 2$ consultas. La tabla de abajo muestra algunos valores de Q y su puntaje asociado.

Q	2	3	4	5	...	20	...	50	...	2500
Puntaje	100	82	71	63	...	32	...	20	...	3

Herramienta de pruebas

Para facilitarte poder probar tu solución, te damos una herramienta simple que puedes descargar. Ve a "attachments" al final de la página del problema en Kattis. El uso de la herramienta es opcional y puedes cambiarla. Ten en cuenta que el evaluador oficial en Kattis es diferente de esta herramienta de pruebas.

Ejemplo de uso (con $H = 4$, $W = 5$, y la caja escondida en la posición $r = 2$, $c = 3$):

Para programas en Python, si tenemos `solution.py` (se corre normalmente `pypy3 solution.py`) corre:

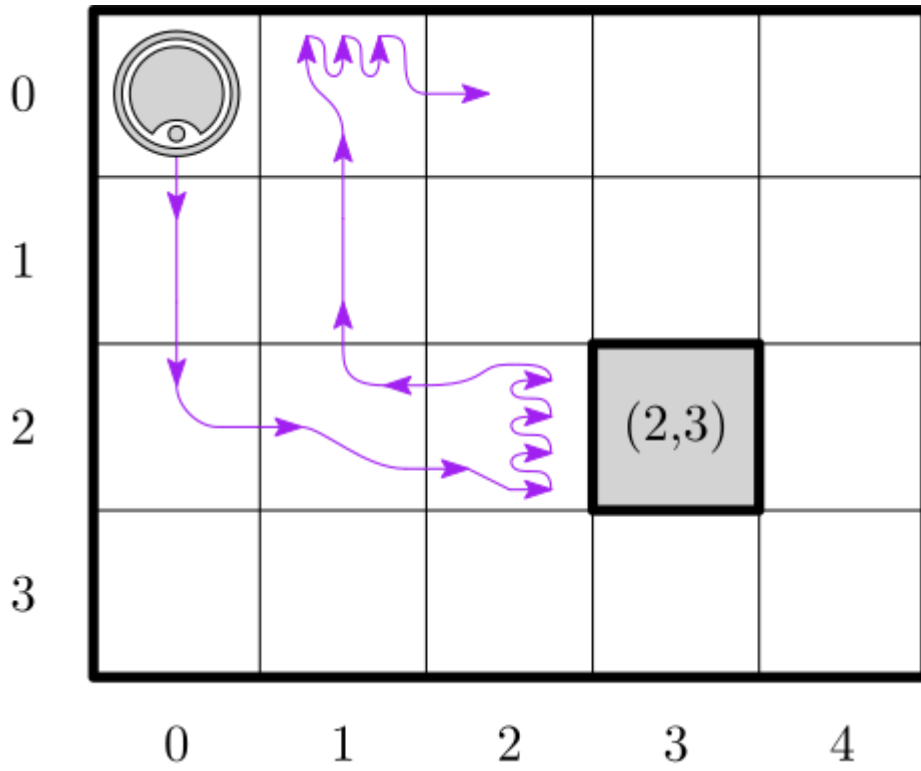
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

Para programas en C++, primero compílalo (por ejemplo `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) y luego corre:

```
python3 testing_tool.py ./solution.out <<<"4 5 2 3"
```

Ejemplo

Considera el caso de prueba de ejemplo. La matriz tiene $H = 4$ y ancho $W = 5$, la caja está en la posición $(r, c) = (2, 3)$. La imagen de abajo muestra el camino que siguió el robot al obedecer las instrucciones de la primer consulta "`? vv>>>>>><^^^^>`", lo que hizo que el robot terminara en la posición $(r, c) = (0, 2)$. Antes de la segunda consulta, el robot regresará a la casilla superior izquierda $(0, 0)$ otra vez. Después la solución hace otra consulta "`? >>>>>>>vvvvvvvvvv`" para la cual el robot termina en la casilla inferior derecha $(r, c) = (3, 4)$. Ahora la solución decide adivinar la respuesta, escribiendo "`! 2 3`", que es la posición correcta de la caja.



Salida del evaluador	Tu salida
4 5	
	? vv>>>>>><^ ^ ^ ^ ^ ^ ^ ^>
0 2	
	? >>>>>>>>vvvvvvvvvv
3 4	
	! 2 3