

Acomoda el número

Tiempo límite por caso: 1 seg
Memoria límite por caso: 256MB

Tienes una **lista desordenada** con **N** números **distintos** los cuales tienen valores entre **0** y **M** . Llamemos al primer número de esta lista **a_0** . Debes ordenar los números de la lista y ver en qué posición queda el número **a_0** .

En este problema la primera posición de la lista es la posición 0 y la última es la posición $N-1$.

Problema

Escribe un programa que dada la lista de **N** números determine en que posición queda el elemento **a_0** (el primero de la lista) después de que ésta se ordena.

Entrada

Tu programa debe leer del teclado la siguiente información:

- En la primera línea el número **N** , la cantidad de elementos en la lista.
- En la segunda línea los **N** números de la lista separados cada uno por un espacio.

Salida

Tu programa debe escribir en la pantalla un único número entero que representa la posición final del elemento **a_0** en la lista ordenada.

Restricciones

$1 < N \leq 2,000,000$

$0 \leq M \leq 10^9$

Cantidad de números en la lista.

Ningún elemento de la lista será menor a **0** ni mayor a **10^9** .

Ejemplo

Entrada	Salida	Explicación
5 14 2 1 17 23	2	Recuerda que las posiciones de la lista se cuentan a partir de 0. La lista ordenada queda 1, 2, 14, 17, 23, de modo que 14, que era el primer elemento en la lista desordenada, quedó en la posición 2 .

Evaluación

Para un grupo de casos con valor de 51 puntos **$N \leq 5,000$** .

Portal

Tiempo límite por caso: 1seg
Memoria límite por caso: 128MB

Probablemente hayas jugado *Portal*, si no lo has hecho, *Portal* es un videojuego en el que el personaje principal recorre diversos laberintos para llegar a su objetivo, un pastel.

El personaje cuenta con un arma capaz de *disparar* portales, es decir, si se apunta con el arma a una pared (**no importa si la pared está retirada, basta con que sea visible desde donde se está**) se *abre* un portal en esa pared. Después de disparar, el jugador puede moverse y el portal seguirá abierto. El arma puede mantener abierto sólo un portal a la vez, de modo que si se utiliza el arma para disparar un nuevo portal, el primero se *cerrará* en el momento que se abre el nuevo.

El jugador cuenta con 2 armas, una dispara portales azules y la otra naranjas. Es sólo usando ambas que los portales son de utilidad. Los portales funcionan así:

- Cuando se dispara un portal a una pared, al llegar a esta pared (el portal puede dispararse desde lejos), el jugador puede *entrar* al portal.
- Si se entra al portal azul, se sale en el cuadro libre junto a la pared donde esté el portal naranja. Si no hay portal naranja abierto, entonces el portal azul no sirve y **no tiene ningún efecto tratar de entrar en él**.
- Viceversa, si se entra al portal naranja, se sale por donde esté el portal azul. Igualmente, si no hay portal azul, entrar al portal naranja no causa ningún efecto.

Para este problema recibirás el mapa de un mundo representado por una cuadrícula rectangular de M filas x N columnas, el mapa marca los cuadros por los que se puede avanzar y los cuadros que son pared, además marca la posición inicial del jugador (x_j, y_j) , y la posición del pastel (x_p, y_p) . Como jugador puedes dar un **paso**, siempre y cuando **no exista** una pared que te lo impida, en las direcciones norte, sur, este u oeste. Además puedes disparar cualquiera de los portales, azul o naranja, igualmente en dirección norte, sur, este u oeste, en caso de disparar un portal este se abrirá en la próxima pared en esa dirección, el mundo está rodeado por pared, de modo que siempre habrá una pared en donde se abra el portal.

Problema

Escribe un programa que, recibiendo el mapa, las coordenadas de inicio del jugador y las coordenadas del pastel, determine cuál es el menor número de pasos que tiene que dar el jugador para llegar hasta el pastel o que imprima 'THE CAKE IS A LIE' si es imposible llegar al pastel.

Entrada

Tu programa debe leer del teclado los siguientes datos:

- La primera línea contiene dos enteros M , N , la cantidad de filas y de columnas en el mapa.
- Las siguientes M líneas tienen N caracteres que representan el mapa, cada uno de estos caracteres puede tener los siguientes valores:
 - '.' (punto): Es un cuadro por donde se puede caminar.
 - '#' (gato): Indica un cuadro por donde no se puede caminar (**pared**).
 - 'O' (o mayúscula): Indica la posición donde inicia el jugador.
 - 'X' (x mayúscula): Indica la posición donde está el pastel.

Salida

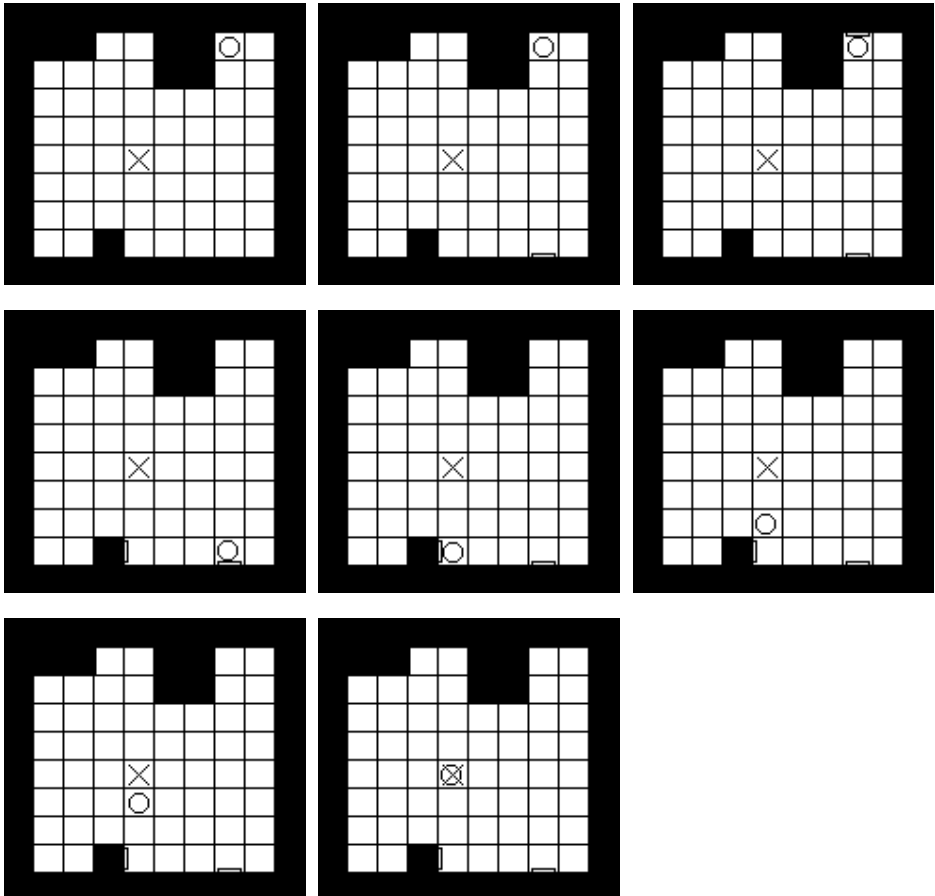
Tu programa debe escribir en la pantalla un único número que indica la cantidad mínima de pasos necesaria para que el jugador llegue al pastel.

Restricciones

$1 \leq M, N \leq 1,250$

Cantidad de filas y de columnas en el mapa

Ejemplo

Entrada	Salida	Explicación
<pre> 10 10 ##### ###..#O.# #...#.# #.....# #.....# #...X...# #.....# #.....# #..#.....# ##### </pre>	5	<p>Para llegar en 4 pasos el jugador tiene que hacer lo siguiente (para que los entiendas sigue los pasos en el mapa de la imagen):</p> <ul style="list-style-type: none"> • Disparar el portal azul hacia el sur. • Disparar el portal naranja hacia el norte. • Avanzar hacia el norte para entrar al portal naranja y salga por la parte de abajo del mapa. Hasta aquí ha dado 1 paso. • Disparar el portal naranja hacia el oeste. • Avanzar hacia el sur para entrar al portal azul y salir por el naranja abajo del pastel. Hasta aquí llevas 2 pasos. • Avanzar 3 pasos al norte para llegar al pastel. En total se dieron 5 pasos. 

Evaluación

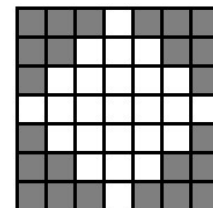
Para un grupo de casos con valor de 21 puntos $M, N \leq 15$ y el camino óptimo al pastel no requiere del uso de portales.
Para un grupo de casos con valor de 34 puntos $M, N \leq 15$ y el camino óptimo al pastel **requiere** del uso de portales.
Para un grupo de casos con valor de 23 puntos $M, N \leq 50$ y el camino óptimo al pastel **requiere** del uso de portales.
Para un grupo de casos con valor de 22 puntos $M, N \leq 1,250$ y el camino óptimo al pastel **requiere** del uso de portales.

Tejidos

Tiempo límite por caso: 1seg
Memoria límite por caso: 64MB

Las familias Escocesas solían identificarse por sus tejidos de Tartán, probablemente los hayas visto, son esos tejidos de cuadros parecidos a los uniformes escolares. Cada familia tenía su propio patrón de tejido.

El COMI ha decidido crear un patrón de tejido para identificar a los miembros de la OMI, pero los cuadros no nos gustan, así que nuestro patrón utilizará rombos. Como somos informáticos, para elegir nuestro patrón, utilizamos un programa de computadora. Este programa generó, mediante un algoritmo secreto, una cuadrícula de $N \times N$ cuadros. Cada uno de ellos puede ser blanco o negro.



Dentro de la cuadrícula quedaron varios rombos blancos sobre **fondo negro** como el que se muestra en la figura. Debes buscar, dentro de la cuadrícula, cuál es el rombo blanco sobre **fondo negro** de mayor tamaño (**no basta con tener un rombo blanco, se requiere además que todos los cuadros en el resto del cuadrado que lo delimita sean negros, como se observa en la figura**). El tamaño del rombo es el largo de su diagonal, para el rombo de la figura su tamaño es 7. **Un punto no se considera rombo, el tamaño mínimo para un rombo es 3.**

Problema

Escribe un programa que recibiendo N , el tamaño de la cuadrícula, y el color de cada uno de los cuadros en ella encuentre el rombo blanco sobre negro de mayor tamaño. En caso de que no exista ningún rombo, tu programa deberá escribir -1

Entrada

Tu programa debe leer del teclado la siguiente información:

- En la primer línea el número N , el tamaño de la cuadrícula.
- En las siguientes N líneas hay N caracteres sin espacios entre ellos. Cada una representa una línea de la cuadrícula, cada carácter representa el color del cuadro correspondiente. Una 'O' (o mayúscula) representa un cuadro blanco, un '.' (punto) representa un cuadro negro.

Salida

Tu programa debe escribir en pantalla un único número entero que indique el tamaño del rombo blanco sobre negro más grande dentro de la cuadrícula.

Restricciones

$1 < N \leq 2,000$ Tamaño de la cuadrícula.

Ejemplo

Entrada	Salida	Imagen	Explicación
<pre> 9 ..O..O... .OOO.OO.O OOOOOO... .OOO..... O.O..O...OOO. O..OOOOO OOO..OOO. .O....O.. </pre>	5		<p>En la imagen puedes observar que hay dos rombos que cumplen con la descripción, uno de tamaño 3 en la esquina inferior izquierda y uno de tamaño 5 en la esquina inferior derecha.</p> <p>El rombo en la parte superior izquierda no es válido porque el fondo no es totalmente negro.</p> <p>Al ser el rombo de tamaño 5 el más grande, la respuesta es 5.</p>

Evaluación

Para un grupo de casos de prueba con un valor de 60 puntos $N \leq 100$.

Secuencias contiguas

Tiempo límite por caso: 1seg
Memoria límite por caso: 64MB

Se tiene una lista **desordenada** de los números entre **1** y **N**.

Llamaremos *secuencia contigua* a un conjunto de elementos de la lista tales que cumplan con las siguientes restricciones:

- Se elige un índice i , un índice j tal que $j > i$ y todos los elementos entre i y j pertenecen al conjunto. Es decir, todos los elementos del conjunto estén contiguos en la lista.
- Cuando se ordenan los elementos del conjunto si el número menor en el conjunto es x y el número mayor en el conjunto es y entonces todos los números enteros z tales que $x < z < y$ también están en el conjunto.

(Te recomendamos leer la explicación del ejemplo para que estas restricciones sean más claras)

El *largo* de una secuencia contigua se define como la cantidad de elementos en el conjunto.

Llamaremos *pareja de secuencias contiguas* a dos secuencias contiguas con el mismo largo y que además estén consecutivas en la lista, es decir, entre la primera secuencia contigua **A** y la segunda **B** no existe ningún elemento.

El *largo* de una pareja de secuencias contiguas es el doble del largo de cualquiera de las secuencias contiguas que la componen.

Problema

Escribe un programa que dada la lista desordenada de los números entre **1** y **N** encuentre el largo de la pareja de secuencias contiguas más larga en la lista.

Entrada

Tu programa debe leer del teclado los siguientes datos:

- La primera línea contiene el entero **N**, la cantidad de elementos en la lista.
- La siguiente línea contiene **N** números enteros separados por un espacio, la lista.

Salida

Tu programa debe escribir en la pantalla un único entero que indique el largo de la pareja de secuencias contiguas más larga en la lista

Restricciones

$1 < N \leq 50,000$ El número de elementos en la lista

Ejemplo

Entrada	Salida	Explicación
10 4 6 1 3 2 9 7 8 5 10	6	El conjunto (1, 3, 2) que están entre los índices 3 y 5 de la lista (asumiendo que la primera posición es el índice 1) forman una secuencia contigua. Todos los elementos entre el índice 3 y el 5 están en el conjunto y además al ordenar el conjunto se cumple que todos los enteros entre el 1 (menor del conjunto) y el 3 (mayor del conjunto) pertenecen al mismo. Igualmente el conjunto (9, 7, 8) forman una secuencia contigua. Ambos conjuntos forman una pareja de secuencias contiguas porque están consecutivos en la lista.

Evaluación

Para un grupo de casos de prueba con valor de **26** puntos, $N \leq 200$.

Para otro grupo de casos de prueba con valor de **29** puntos, $N \leq 1,000$.

Para el último grupo de casos de prueba con valor de **45** puntos, $N \leq 50,000$.