



Nowruz

Faltan pocos días para Nowruz (el año nuevo Persa), y el abuelo ha invitado a la familia a su Jardín. Hay k niños entre los invitados. Para hacer la reunión más divertida para los niños, el abuelo va a organizar un juego de escondidas.

El Jardín puede ser representado por una grilla de $m \times n$ casillas. Algunas casillas (posiblemente cero) están bloqueadas por rocas, y las restantes las llamamos *libres*. Dos casillas son *vecinas* si comparten un lado. Esto es, cada casilla tiene hasta 4 vecinos: dos en la dirección horizontal, y dos en la dirección vertical. El abuelo quiere convertir su jardín en un laberinto. Para este propósito, él puede bloquear algunas casillas libres plantando arbustos en ellas. Las casillas donde el planta los arbustos dejan de ser libres.

Un laberinto debe tener la siguiente propiedad. Para cada par a y b de casillas libres en el laberinto debe existir exactamente un *camino simple* entre ellas. Un camino simple entre las casillas a y b es una secuencia de casillas libres que empieza en a , termina en b , todas las casillas son distintas, y cada par de casillas consecutivas son vecinas.

Un niño puede esconderse en una casilla si y solo si esa casilla es libre y tiene *exactamente* un vecino libre. Dos niños no se pueden esconder en la misma casilla.

El mapa del jardín es dado como entrada. Tu tarea es ayudar al abuelo a crear un laberinto en el que muchos niños se puedan esconder.

Detalles de implementación

Esta es una tarea *output-only* con puntaje parcial. Hay 10 archivos de entrada, cada uno describiendo el jardín del abuelo. Para cada archivo de entrada debes enviar un archivo de salida con el mapa del laberinto. Por cada archivo de salida recibirás puntos basados en el número de niños que se pueden esconder en tu laberinto.

No debes enviar ningún código para esta tarea.

Formato de entrada

Cada archivo de entrada describe una grilla representando un jardín y da el número k de niños invitados por el abuelo. El formato es como sigue:

- Línea 1: m n k
- Línea $1 + i$ (para $1 \leq i \leq m$): fila i de la grilla, que es una cadena de longitud n , compuesta por los siguientes caracteres (sin espacios):

- '.': una casilla vacía,
- '#': una roca.

Formato de salida

- Línea i (para $1 \leq i \leq m$): fila i del laberinto (el jardín, después de plantados los arbustos). Es una cadena de longitud n , compuesta de los siguientes caracteres (sin ningún espacio):
 - '.': una casilla vacía,
 - '#': una roca,
 - 'X': un arbusto. (Note que la letra X debe ser mayúscula)

Restricciones

- $1 \leq m, n \leq 1024$

Puntuación

Un archivo de salida es considerado *válido* si se cumplen todas las siguientes condiciones:

- El mapa de la salida debe coincidir con el mapa de la entrada, con la única excepción que un número arbitrario de caracteres '.' en la entrada pueden ser cambiados por caracteres 'X' (celdas bloqueadas por arbustos).
- El mapa de salida debe ser un laberinto, según se define en el enunciado.

Si tu salida para un caso de prueba no es válida, tu puntaje para ese caso será 0. De otra manera, el puntaje será $\min(10, 10 \cdot l/k)$ puntos, redondeado hacia abajo dos dígitos después del punto decimal. Aquí, l es el número de niños que se pueden esconder en tu laberinto, y k el número dado en la entrada. Obtendrás 10 puntos para un caso de prueba si y solo si tu salida es un laberinto en el que k o más niños pueden esconderse. Para cada caso de prueba existe una solución que obtiene 10 puntos.

Nota que si tu solución es válida pero obtiene 0 puntos de acuerdo a la formula de arriba, el sistema de calificación mostrará 'Wrong Answer' como veredicto en el CMS.

Ejemplo

Considera la siguiente entrada:

```
4 5 5
....#
.#..#
...#.
....#
```

Abajo se muestra una posible salida:

```
.X.X#  
.#..#  
...#X  
XX..#
```

Como $l = 4$ niños pueden esconderse en este laberinto, esta solución obtendrá $10 \cdot 4/5 = 8$ puntos. Las celdas en las cuales los niños pueden esconderse están marcadas con \circ a continuación:

```
OXOX#  
.#.O#  
...#X  
XX.O#
```

Las siguientes tres salidas no son válidas:

```
.XXX#      ...X#      XXXX#  
.#XX#      .#.X#      X#XX#  
...#.      ...#X      ..X#X  
XX..#      XXXX#      ..XX#
```

En la salida de la izquierda no hay camino simple entre la celda libre de la esquina superior izquierda y la celda libre de la columna de más a la derecha. En las otras dos salidas, para cada par de celdas libres distintas hay exactamente dos caminos simples distintos entre ellas.