

---

## Laberinto (4): DFS

**P64856\_es**

Se te da el mapa de un laberinto, donde las casillas marcadas con un ‘#’ son paredes, y las marcadas con un ‘.’ son espacios vacíos. Partiendo de una casilla inicial vacía ‘A’, se te pide que te muevas por el laberinto siguiendo el siguiente proceso, conocido como DFS (Depth-First Search, o búsqueda en profundidad):

- ‘AVANZAR’: Siempre que puedes, te mueves hacia arriba (norte); si la casilla a la que te moverías es una pared, o emphaya la has visitado con anterioridad, entonces te mueves hacia la derecha (este); si esto no es posible (por cualquiera de los dos motivos anteriores) te mueves hacia abajo (sur); finalmente, si tampoco es posible, te mueves hacia la izquierda (oeste).
- ‘RETROCEDER’: Si no puedes avanzar hacia ninguna dirección, entonces retrocedes sobre tus pasos tantas veces como sea necesario (incluyendo, si es necesario, hasta la casilla inicial) hasta que sea posible volver a avanzar.

Se te pide que realices el proceso anterior sobre el mapa y que marques, para cada casilla (incluyendo la inicial), el número de pasos que llevas realizados cuando la visitas por primera vez (incluyendo los pasos en que retrocedes).

Para hacerlo, te resultará útil hacer un programa recursivo o bien usar una estructura de datos como una pila (‘stack’), donde puedes “apilar” los pasos que avanzas y “desapilar” cada paso que retrocedes.

### Entrada

La entrada consiste de una línea con un número  $k \geq 0$ , seguido de  $k$  casos. Cada caso empieza con una línea con las dimensiones  $n$  (filas) y  $m$  (columnas) del mapa, seguida de  $n$  filas de  $m$  caracteres ‘#’ y ‘.’ con la descripción del mapa, y un único carácter ‘A’ con la posición inicial.

### Salida

Para cada caso, escribe el mapa usando exactamente  $d$  caracteres para cada casilla, donde  $d$  es el mínimo número de dígitos necesario para representar todos los números de pasos que acabarás mostrando (por ejemplo, si alcanzas la última casilla después de 103 pasos, entonces  $d = 3$ ). Si la casilla es una pared, escribe ‘x’  $d$  veces; si la casilla está vacía pero no es posible llegar a ella, escribe ‘.’  $d$  veces; de otro modo, escribe el número de pasos necesarios para llegar a ella usando exactamente  $d$  dígitos (añade ceros a la izquierda si es necesario). Por último, escribe un espacio entre cada par de casillas de la misma fila. Fíjate en los ejemplos para salir de dudas.

Separa dos casos de pruebas con una línea con 3 asteriscos (‘\*\*\*’).

### Puntuación

- **TestA:**

**50 Puntos**

Entradas con  $k \leq 100$  y  $n, m \leq 10$ , y donde todas las casillas situadas en los bordes del mapa son paredes (‘#’), como el Ejemplo 1.

● **TestB:**

**50 Puntos**

Entradas con  $k \leq 10$  y  $n, m \leq 200$ , con mapas de todo tipo, como el Ejemplo 2.

### Ejemplo de entrada 1

```
5
6 6
######
#.#
#.A.#
###.##
...##
#####
5 9
##########
.....#
##....#
#.A#...
######
4 9
##########
...#####
##...A#
######
7 9
######
....#
...#
...#
....#
...#
.A#...
######
5 10
######
..#...#...
#.##.A#...
....#
######
```

### Ejemplo de salida 1

```
## ## ## ## ## ##
## .. ## 01 02 ##
## .. ## 00 03 ##
## ## ## 07 ##
## 10 09 08 ##
## ## ## ## ## ##
***  
## ## ## ## ## ## ## ## ## ##
## 25 24 19 02 03 04 05 ##
## ## ## 20 01 10 09 06 ##
## .. ## 21 00 ## 08 07 ##
## ## ## ## ## ## ## ## ##
***  
# # # # # # # #
# . . # # # # #
# # # . . . # 0 #
# # # # # # # # #
***
```

## Ejemplo de entrada 2

## Ejemplo de salida 2

## Información del problema

Autoría: Omer Giménez

Generación: 2026-01-25T11:21:59.503Z

© Jutge.org, 2006–2026.  
<https://jutge.org>