

---

**Laberinto (5): BFS****P33254\_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 explores por el laberinto siguiendo el siguiente proceso, conocido como BFS (Breadth-First Search, o búsqueda en anchura):

- Partiendo de la casilla inicial, avanza un paso en cualquier dirección. Las, como mucho, cuatro casillas vacías que encuentres son las casillas que están a 1 paso de distancia.
- Partiendo de cualquiera de las casillas que están a 1 paso de distancia, avanza un paso en cualquier dirección posible. Las casillas vacías que encuentres *que no has visitado con anterioridad* son las casillas que están a 2 pasos de distancia.
- En general, partiendo de cualquiera de las casillas que están a  $n$  pasos de distancia y avanzando un paso en cualquier dirección posible, las casillas vacías que encuentres *que no has visitado con anterioridad* son las casillas que están a  $n + 1$  pasos de distancia.

Se te pide que hagas un programa que encuentre a qué distancia en pasos está cada casilla. Para hacerlo, te resultará útil utilizar una estructura de datos como una cola ('queue'), donde puedes acumular las casillas que encuentres a distancia  $n + 1$  mientras estudias las casillas que están a distancia  $n$ .

**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 posibles (por ejemplo, si la casilla más lejana a la que llegas está a 103 pasos de distancia, 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
5 6
#####
#....#
#.#A.#
###.##
#####
5 10
#####
#.....#
###.....#
#.#.A#...#
#####
4 9
#####
#..#####
###...#A#
#####
7 9
#####
#....#..#
#..#...#.#
#...#...#
#....#...#
#.A.#...#
#####
5 10
#####
#.#..#..###
#...#.A#...#
#...#...##
#####
```

## Ejemplo de salida 1

```
# # # # #
# 3 2 1 2 #
# 4 # 0 1 #
# # # 1 # #
# # # # #
***
# # # # # # # # #
# 5 4 3 2 3 4 5 6 #
# # # 2 1 2 3 4 5 #
# . # 1 0 # 4 5 6 #
# # # # # # # # #
***
# # # # # # # # #
# . . # # # # # # #
# # # . . # 0 #
# # # # # # # # #
***
## ## ## ## ## ## ## ## ##
## 05 04 05 06 ## 14 13 ##
## 04 03 ## 07 08 ## 12 ##
## 03 02 03 ## 09 10 11 ##
## 02 01 02 03 ## 11 12 ##
## 01 00 01 ## 13 12 13 ##
## ## ## ## ## ## ## ## ##
***
# # # # # # # # #
# . # 3 2 # . # # #
# . # # 1 0 # 4 5 #
# . . . # 1 2 3 # #
# # # # # # # # #
```

## Ejemplo de entrada 2

```
1
9 18
....#...#.....
...#...###...###.
....#####...#.#.
...#.....###.##.
..#....#...#...#
..#.A...#...#...
..#.....##.....#
#..#.#.....#...#..
....#.....###...
```

## Ejemplo de salida 2

```
.. .. .. ## .. .. ## .. .. .. .. .. .. .. ..
.. .. .. ## .. .. .. .. ## ## ## .. .. .. ## ## ## ..
.. .. .. ## ## ## ## 07 08 09 ## ## .. .. .. ## ..
.. .. .. ## 02 03 04 05 06 07 08 ## ## .. .. ## ## ..
.. .. ## 02 01 02 03 ## 07 08 09 ## .. .. ## 16 17 ##
.. .. ## 01 00 01 02 ## 06 07 08 09 ## ## 14 15 16 17
.. .. ## 02 01 02 03 04 05 ## ## 10 11 12 13 14 ## 18
## .. .. ## 02 ## 04 05 06 07 08 ## 12 13 14 ## 20 19
.. .. .. .. ## 06 05 06 07 08 09 10 ## ## ## 22 21 20
```

## Información del problema

Autoría: Omer Giménez

Generación: 2026-01-25T10:17:27.172Z

© Jutge.org, 2006–2026.

<https://jutge.org>