

# Fundamentos de Programación

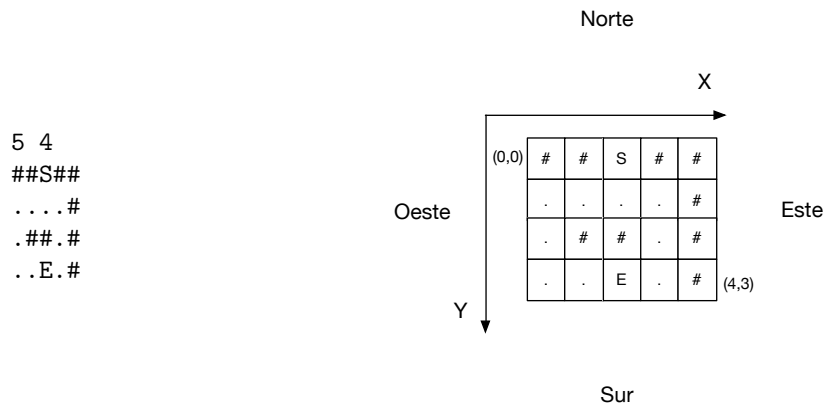
## Tarea 5

Luego de su aterrizaje en Marte el año 2012, el robot explorador *Curiosity* ha descubierto un intrincado sistema de cuevas—que parecen un laberinto diseñado por una extraña civilización—en el planeta rojo. Lamentablemente, los sensores del robot están bastante deteriorados y no es posible la comunicación subterránea para darle nuevas instrucciones. A pesar de que los científicos de la NASA han logrado determinar que dichos laberintos tienen marcadores químicos que indican la entrada y salida de los laberintos, ellos necesitan su ayuda para programar al robot *Curiosity* con un algoritmo recursivo de forma que pueda descubrir algún camino para llegar desde la entrada a la salida.

**Se le entregará una plantilla de la solución donde usted sólo debe implementar la función de búsqueda recursiva encontrarCamino. Puede modificar esta plantilla y agregar otras funciones si así lo requiere!**

### Entrada

La entrada consiste en la descripción del laberinto como una matriz de caracteres. La primera línea contiene 2 enteros,  $n$  y  $m$ , que denotan la cantidad de columnas y filas, respectivamente. Luego, vienen  $m$  líneas, cada una con  $n$  caracteres. Considere que  $2 \leq n, m \leq 10$ . Por ejemplo, a continuación se muestra a la izquierda la entrada como texto, y a la derecha su interpretación como el mapa del laberinto:



Observe que la posición inicial del robot se muestra con la letra E, y la posición final del laberinto se muestra con la letra S. Los # indican obstáculos o muros, por los cuales el robot **no puede avanzar**, y los . indican lugares por donde el robot sí puede avanzar. Es importante destacar que los valores de la

coordenada Y están “al revés”, es decir, mientras más hacia “abajo”, mayor es el valor de Y, como ejemplos se puede observar la coordenada origen (0,0) y la coordenada inferior derecha (4,3). Todas las descripciones de laberinto tendrán marcado sólo una posición inicial y sólo una posición final; todos los laberintos tendrán la misma orientación respecto a los puntos cardinales.

**Movimiento del Robot** Debido a los daños y a la estructura de los laberintos, el robot sólo puede moverse hacia el Norte, Sur, Este, u Oeste. Es decir, **no se puede mover en diagonal** dentro del laberinto. Además, el robot sólo debiera moverse de una casilla a la vez (esto le ayudará a pensar su solución recursiva!).

## Salida

La salida del programa depende de si existe un camino desde la posición inicial hasta la posición final del laberinto. Si existe un camino, debe imprimir por pantalla el mapa marcando cada paso del camino con un signo +. En caso que exista más de un camino, cualquiera es válido. Si no existe un camino, debe imprimir “No hay camino.”. Por ejemplo:

Entrada	Salida
5 4	
##S##	##S##
....#	+++.#
.##.#	+++.#
..E.#	+++.#
Entrada	Salida
3 4	No hay camino.
##S	
###	
...	
..E	

## Solución Recursiva

La clave para la solución recursiva es descomponer el problema general, *encontrar un camino hasta la salida*, en problemas más pequeños, es decir, en muchas búsquedas de caminos, donde el robot avanza sólo una casilla. Veamos el mapa de ejemplo:

```
##S##
....#
.##.#
..E.#
```

donde la posición inicial es (2, 3), la cual es parte del camino pedido, si es que existe. Para reducir el problema debemos ver si existe un camino entre una nueva posición (x,y), a la cual nos podemos mover desde (2,3), y la salida del laberinto. Si es así, tenemos un camino entre la posición inicial y la posición final, puesto que sabemos cómo ir desde (2,3) hacia (x,y). Los movimientos posibles son hacia el Norte, Sur, Este u Oeste. No obstante, observe que no todos los movimientos son válidos, por ejemplo el robot no puede pasar por casillas

con obstáculos, o por coordenadas que no existen en el mapa del laberinto. La idea de la solución recursiva es probar cada camino posible mediante llamadas recursivas. Es decir, se debe utilizar la recursividad para generar una enumeración exhaustiva de los posibles movimientos del robot, teniendo en cuenta si la búsqueda por un camino es exitosa o no.

En los casos en que no se encuentra un camino hacia la salida este debe descartarse y luego el algoritmo debe realizar una “vuelta hacia atrás” (una técnica conocida como *backtracking recursivo*) para seguir probando las otras posibilidades. Esto significa que el robot puede ir y devolverse varias veces hasta encontrar un camino. Si se descartan todos los caminos posibles, entonces no existe camino hacia la salida. Como ejemplo, considere un conjunto posible de movimientos para encontrar la solución al mapa de ejemplo:

Leer de arriba hacia abajo, y de izquierda a derecha...	10) Mov: 0, 2 Ya visitado	
Mapa Laberinto	5) Mov: 0, 3	11) Mov: 1, 1
##S##	##S##	##S##
....#	....#	++..#
..##.	..##.	+++.#
..E.#	+++.#	+++.#
1) Mov: 2, 3	6) Mov: 0, 2	12) Mov: 1, 0
##S##	##S##	Obstaculo
....#	....#	
..##.	+++.#	13) Mov: 0, 1
..+.#	+++.#	Ya visitado
2) Mov: 2, 2	7) Mov: 0, 1	14) Mov: 1, 2
Obstaculo	##S##	Obstaculo
	+...#	
3) Mov: 1, 3	+++.#	15) Mov: 2, 1
##S##	+++.#	##S##
....#		+++.#
..##.	8) Mov: 0, 0	+++.#
..+.#	Obstaculo	+++.#
4) Mov: 1, 2	9) Mov: -1, 1	16) Mov: 2, 0
Obstaculo	Fuera del laberinto	Llegamos a la meta!

**Ayuda:** Para ver el ruteo que realiza la implementación de referencia, ingrese el mapa en la página <http://zeus.inf.ucv.cl/~ifigueroa/teaching/robot/>.

## Reglas del Juego

- La tarea debe programarse en el lenguaje C. No utilice funciones que no pertenecen al ANSI C.
- Puede desarrollar la tarea usando CodeBlocks, DevCpp, u otro entorno de desarrollo. Sin embargo su tarea debe compilar correctamente en CodeBlocks.
- Los datos de entrada deben proporcionarse mediante teclado y la salida hacia pantalla. Para los casos de prueba se utilizará la redirección desde archivos.
- Se debe respetar las especificaciones entregadas para la entrada y salida.
- La corrección incluye una interrogación.
- Solo se recibirán tareas fuera de plazo dentro de las 24 horas siguientes a la fecha de entrega. Nota máxima es un 5.0.
- **Se considerará una bonificación de hasta 10 puntos por seguir la guía de buenas prácticas que se subirá al Aula Virtual.**

Fecha de entrega : Miércoles 10 de Junio. Enviar código fuente `tarea5.c`, indicando en su interior nombre y rut, por email hasta las 24:00. Los correos según código y paralelo son:

- ICI-1142-1: `ici1142_1_s12015@inf.ucv.cl`
- INF-1142-1: `inf1142_1_s12015@inf.ucv.c`
- INF-1142-2: `inf1142_2_s12015@inf.ucv.c`