

Fundamentos de Programación

Tarea 2

¿Has jugado alguna vez *Buscaminas*? Este bonito juego viene instalado en cierto sistema operativo cuyo nombre no podemos recordar. La meta del juego es encontrar la ubicación de todas las minas dispersas dentro de un campo de $M \times N$ casillas.

Al comienzo del juego todas las casillas están ocultas. Al seleccionar una casilla ésta puede contener una mina, lo que te hace perder una vida. Si no, la casilla muestra un número que indica cuántas minas hay adyacentes a esa casilla. Cada casilla tiene a lo más ocho casillas adyacentes. Por ejemplo, el campo de 4×4 que se muestra a continuación, a la izquierda, tiene 2 minas, cada una representada por un asterisco “*”. Si representamos el mismo campo con los números según lo descrito anteriormente, se obtiene el campo que está a la derecha:

*...	*100
....	2210
.*..	1*10
....	1110

Entrada

La entrada consiste en la descripción de un campo minado. La primera línea de la entrada contiene dos enteros n y m , que corresponden a la cantidad de filas y columnas del campo, respectivamente. Considere que $0 < n, m \leq 100$. Luego cada una de las siguientes n líneas contiene exactamente m caracteres representando el campo. Las casillas seguras, que no tienen una mina, están representadas por un punto “.” y las casillas que poseen una mina están representadas por un asterisco “*” (ambos sin las comillas).

Salida

La salida consiste en la descripción del campo minado con los números que indican las minas adyacentes a cada casilla. Específicamente, la salida contiene n líneas donde los caracteres “.” de la entrada son reemplazados por la cantidad de minas adyacentes a dicha casilla. Ejemplos:

Entrada	Salida
4 4	
*...	*100
....	2210
.*..	1*10
....	1110

Entrada	Salida
3 5	
**...	**100
.....	33200
.*...	1*100

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 : Martes 14 de Abril. Enviar código fuente `tarea2.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`