



El juego del Buscaminas consiste en descubrir las minas ocultas en un tablero de $M \times N$ casillas a partir de las indicaciones que nos da el propio juego acerca del número de minas que se ocultan en las casillas adyacentes a una casilla dada. El ejercicio de programación aquí propuesto parte de esa misma estructura –el campo de minas representado por una matriz $M \times N$ – para ofrecer como resultado un nuevo tablero con las minas descubiertas y el número de minas adyacentes para cada casilla que no tenga mina.

Considérese el siguiente tablero de ejemplo, de 4×4 casillas, donde las minas se representan por un asterisco '*' y las casillas vacías por un punto '.':

```
* . . .
. . . .
. * . .
. . . .
```

Para este caso concreto, la solución que debería entregar el programa sería la siguiente:

```
*100
2210
1*10
1110
```

Se pide escribir un programa capaz de calcular este tipo de soluciones para cualquier tablero de dimensiones arbitrarias. A continuación se describen el formato de entrada de los datos y el formato de salida que deberá respetar el programa. Finalmente se dan algunas indicaciones para su implementación

El formato de entrada de los datos tiene siempre la siguiente estructura: un par de números enteros –separados por espacios– que indican el número de filas y el número de columnas (respectivamente) del

tablero seguidos por tantas líneas como filas haya. Cada línea contiene asteriscos o puntos, tantos como columnas, para codificar si en una casilla hay una mina (asterisco) o no (punto).

A modo de ejemplo, el tablero mostrado anteriormente se codificaría así:

```
4 4
* . . .
. . . .
. * . .
. . . .
```

El programa se debe poder ejecutar de ésta manera

```
$ my_program prueba1.txt
```

```
*100
2210
1*10
1110
```

Lineamientos:

- Usar pruebas unitarias.
- Usar un repositorio de código público.
- Que el repositorio muestre cada avance significativo en la solución.