Compulsory Task 1

Create a file named **minesweeper.py**

Create a function that takes a grid of # and -, where each hash (#) represents a mine and each dash (-) represents a mine-free spot.

Return a grid, where each dash is replaced by a digit, indicating the number of mines immediately adjacent to the spot i.e. (horizontally, vertically, and diagonally).

Example of an input:

```
[ ["-", "-", "-", "#", "#"],
  ["-", "#", "-", "-", "-"],
  ["-", "-", "#", "-", "-"],
  ["-", "#", "#", "-", "-"],
  ["-", "-", "-", "-", "-"] ]
```

Example of the expected output:

```
[ ["1", "1", "2", "#", "#"],
  ["1", "#", "3", "3", "2"],
  ["2", "4", "#", "2", "0"],
  ["1", "#", "#", "2", "0"],
  ["1", "2", "2", "1", "0"] ]
```

Here is a tip. When checking adjacent positions to a specific position in the grid, the following table might assist you in determining adjacent indexes:

NW position = current_row - 1 current_col - 1	N position = current_row - 1 current_col	NE position = current_row - 1 current_col + 1
W position = current_row current_col - 1	Current position = current_row current_col	E position = current_row current_col + 1
SW position = Current_row + 1 current_col - 1	S position = current_row +1 current_col	SE position = current_row + 1 current_col + 1

Also ensure that when checking adjacent positions in the grid that you take into account that on the edges of the grid, you may go out of bounds.

Lastly, you could make use of the enumerate function in Python to keep track of the index points and values without having to create a count variable and explicitly iterate the count variable to keep track of the current row or column index.