

# PE06: Programming Exercise

## Instructions

1. [island\\_counter\\_main.py](#)
2. [island\\_counter.py](#)

## Description:

This assignment is to gain knowledge on the Breadth-first search using a queue. Given a 2D grid map of '1's (land) and '0's (water), count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water. Note that **"island\_counter\_main.py" with the "main" method and starter file "island\_counter.py" with class has already been provided (download attachment).**

As part of the assignment, describe how this problem can be solved using a recursive function instead of a queue. Keep in mind to always comment and document your class and methods.

## Documentation reference:

- Mertz, J. (n.d.). Documenting Python Code: A Complete Guide. <https://realpython.com/documenting-python-code/>

## Expected result:

### *island\_counter\_main.py:*

- This is the Main python file that is already provided and contains the main and test procedure, which calls methods implemented on "islandcounter.py" (this is already provided, but please include this file in your submission).

### *island\_counter.py*

- This class contains such methods as init, print\_grid\_visited, count\_island, and mark\_visited\_island. Please keep in mind the following notes for each method during implementation:
  - init(): this method has already been provided.
  - print\_grid\_visited(): this method has already been provided.
  - count\_island(): this method returns the number of islands while marking already visited grounds.
  - mark\_visited\_island (x, y): this method marks the current location as visited first. After marking the current position, the method looks top/bottom/left/right first to see if it is a valid land and has never been visited **using a queue**. If the island has never been visited, the method marks the location as visited, and the function runs until the queue does not contain any coordinates. **Parameters:** x, y coordinate of the current location

## Writeup

As part of the assignment, describe how this problem can be solved using a recursive function instead of a queue.