Assignment 5: 2D Lists

Winter Code Camp

DUE: January 6, 2021, 11:59PM

At the beginning of each python program, please include comments indicating your name and your student number

Example:

```
# Animesh Sarker
# 105184718
```

Please submit this assignment as a **single zip file** called `FIRSTNAME_LASTNAME_assignment5.zip`.

---

1. Make two helper functions called `empty()` and `find()`. The `empty()` function will take in a 2D list and will return `True` if all values in the 2D list are `0` and `False` otherwise. The `find()` function will take in a 2D list and a value to search for. If the value is found within the 2D list, return a list of length 2 containing the x and y indices of the position of said value. Othwise, return `[-1, -1]` [25 marks]

Sample input:

```
grid = [[23, 34, 67], [44, 5, 3], [7, 8, 9]]
```

Sample output:

```
empty(grid) -> False
find(grid, 67) -> [0, 2]
```

Sample input:

```
grid = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
```

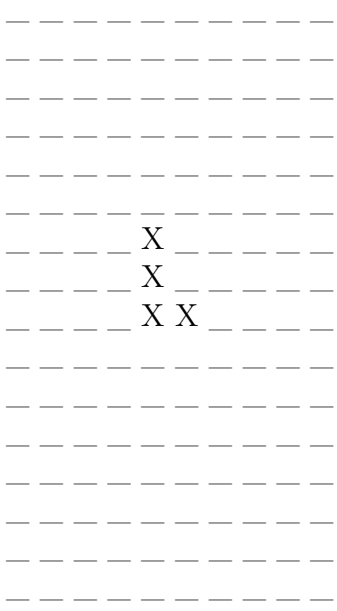
Sample output:

```
empty(grid) -> True
find(grid, 69) -> [-1, -1]
```

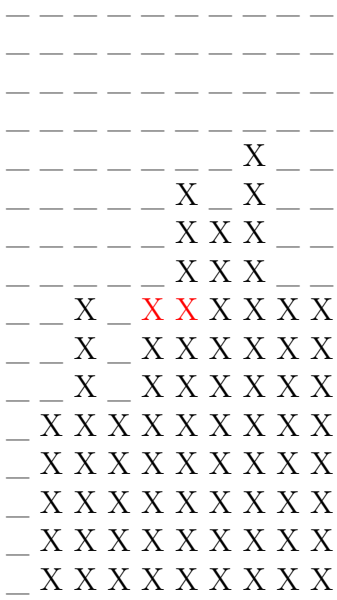
Save file as 2dlistAssign1.py

2. **In the game of Tetris there are a variety of ways to keep track of the piece that is currently falling and the pieces that already fell. One simple approach is to use two 2D lists. One list stores the falling piece, and one stores the blocks on the ground. Each list is 10 x 16 and has either a 0 or 1 in it, where 0 is an empty spot and 1 is a spot that has a block in it. Make a function called `collide` that takes both of these lists as parameters and returns True if there is a collision, and False otherwise. A collision occurs when both grids have a 1 at the same spot. [25 marks]**

Here is an example:



This is the falling piece grid

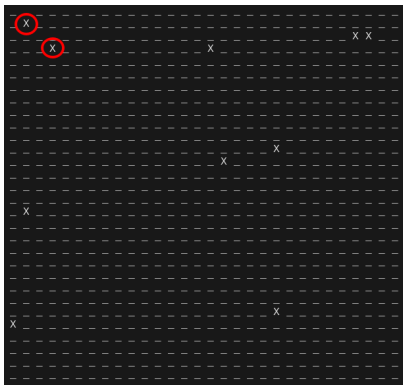


This is the grid containing the pieces on the ground. The red spots indicate where the grids collide, therefore returning True.

Save file as 2dlistAssign2.py

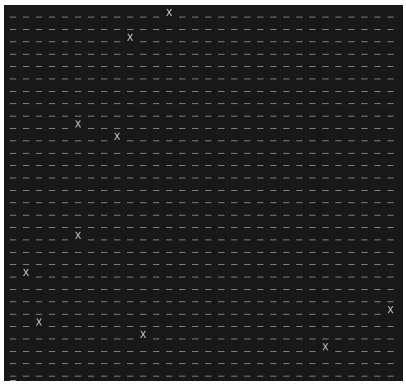
3. **I have a perfectly square 30m x 30m backyard. I bought 10 trees and I want to plant them in totally random places so that it looks more natural. I'm concerned that if we plant any two trees within 3m of each other, then they will have a hard time growing.**
Create a program that makes a 30x30 2D list, and randomly assigns positions for 10 trees. You must keep re-doing the process until no two trees are within 3 meters of each other. [50 marks]
Note: use pythagorean theorem to calculate distance.

For example, here is an invalid backyard:



This backyard is invalid because the two circled trees are 2.82 meters apart. One tree is at position $(1, 1)$ and the other is at $(3, 3)$. The distance is:
$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(3-1)^2 + (3-1)^2} = \sqrt{2^2 + 2^2}$
$= \sqrt{8} \simeq 2.82$ meters

Here is a valid backyard:



Save file as 2dlistAssign3.py