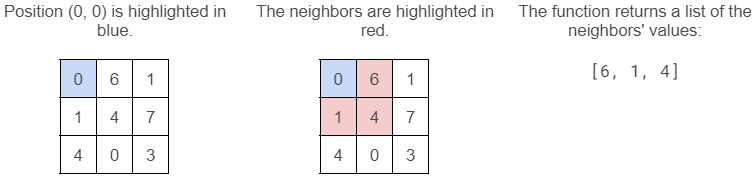
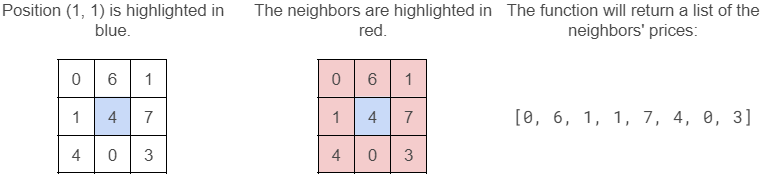
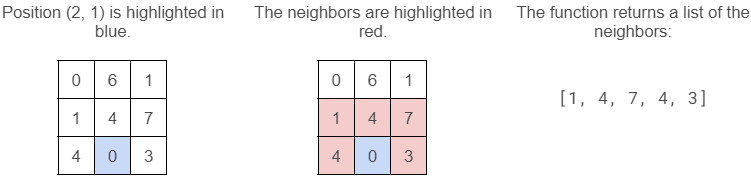
# **Sim City Land Value Calculator Version 2: Find neighbours**

Let’s find our neighbors! Neighbors are all the cells that are in the grid and are horizontally, vertically or diagonally adjacent.

Here is some visuals for *what* is a neighbor:   

## **What to do**

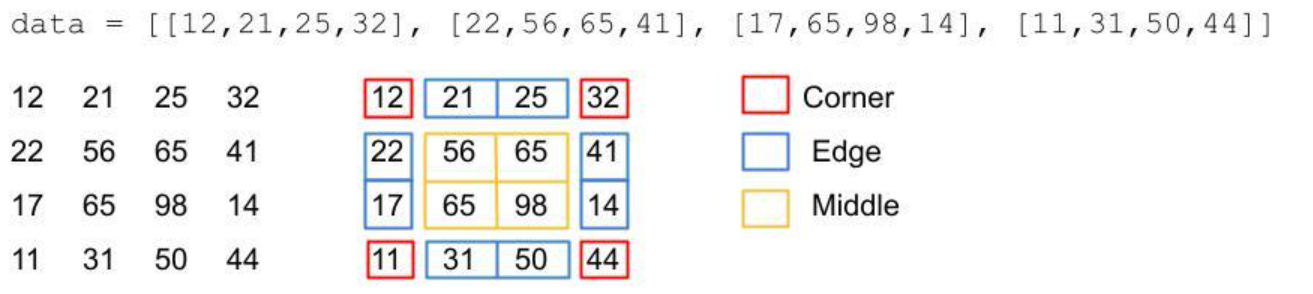
Implement the find\_neighbor\_values() function that finds the neighbours given the coordinate location (row, col).

You must use the following template:

| **def** **create\_grid**(filename: str) -> list[list[int]]:  """  Create a grid of land values from a file  """  *# Implemented in Version 1*  **def** **display\_grid**(grid: list[list[int]]) -> **None**:  """  Display a grid of land values  """  *# Implemented in Version 1*    **def** **find\_neighbor\_values**(grid: list[list[int]], row: int, col: int) -> list[int]:  """  Find the neighbors of a cell  """  *# TODO: Implement this function*  **def** **main**() -> **None**:  """  Main program.  """  grid = create\_grid("data\_0.txt")  print("Sim City Land Values:")  display\_grid(grid) |
| --- |

## **Hints**

* For the find\_neighbor\_values() function, consider three cases:
  + The cell is at the edge of the grid.
  + The cell is in the middle of the grid.
  + The cell is at the corner of the grid. Example:



## **Program name**

Save your program as simcity2.py.

## **Demo**

In this demo, data\_1.txt is used.

<https://asciinema.org/a/Q2LeLXvi5gRnQ6HqXd6IJYMN8>

## **Testing**

To make sure your program works correctly, you should test it.

Good news: we wrote the unit tests for you: [test\_simcity2.py](https://drive.google.com/file/d/1jXh1nzMbCYLqMdAnliZlDcssuOyerPep/view?usp=sharing)

To test your find\_neighbor\_values() function, simply run the unit tests:

| $ python -m pytest test\_simcity2.py |
| --- |

All tests should pass.

## **Submitting**

Submit simcity2.py via eClass.

**Copyright**

I. Akhmetov, J. Schaeffer, M. Morris and S. Ahmed, Department of Computing Science, Faculty of Science, University of Alberta (2022).