

# SimCity Land Value Calculator Version 4:

## Calculate statistics

---

Finally, let's calculate the maximum and average values in the grid, then display these stats.

### What to do

---

- Implement `find_max` that finds the maximum values in the grid and then returns the maximum value.
- Implement `find_average` that finds the average of all the values in the grid and then returns the average.
- Update the `main` function to include calls to `find_max` and `find_average` functions and to print the values returned by these functions.
- You must use the given template means you should not change the names of the functions, its parameters or the object it returns.

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
    """
    # Implemented in Version 2

def fill_gaps(grid: list[list[int]]) -> list[list[int]]:
    """
```

```

    Fill the gaps in the grid
    Creates a new grid that is a copy of the original grid
    Call find_neighbor_values function to find the neighbors of each cell.
    Find the average of their values and round it to the nearest integer
    Use the average values to fill in the missing values in the new grid.
    Return the new grid
    Do NOT modify the original grid!
    """

    # Implemented in Version 3

def find_max(grid: list[list[int]]) -> int:
    """
    Find the max value in the grid (rounded to the nearest integer)
    """
    # TODO: Get the maximum value in the grid
    pass

def find_average(grid: list[list[int]]) -> int:
    """
    Find the average value in the grid (rounded to the nearest integer)
    """
    # TODO: Get the average value of the grid
    pass

def main() -> None:
    """
    Main program.
    """
    grid = create_grid("data_0.txt")
    print("Sim City Land Values:")
    display_grid(grid)
    print("\nCalculated SimCity land values:")
    new_grid = fill_gaps(grid)
    display_grid(new_grid)
    print("\nSTATS")
    print(f"Average land value in this city: {find_average(new_grid)}")
    print(f"Maximum land value in this city: {find_max(new_grid)}")

if __name__ == "__main__":
    main()

```

## Hints

- You can use nested loops to find the maximum and average values.

## Program name

---

Save your program as `simcity4.py`.

## Demo

---

In this demo, `data_1.txt` is used.

<https://asciinema.org/a/NdueVP9SBQfVMQyNhAORMy8QI>

## Testing

---

To make sure your program works correctly, you should test it using different input files found in the introduction document.

- Run your program with `python simcity4.py` with `data_0.txt` Your program should print:

```
Sim City Land Values:
```

```
  1      0      3      4
  5      6      7      8
  9     10     11     12
 13     14     15     16
```

```
Calculated Sim City land values:
```

```
  1      4      3      4
  5      6      7      8
  9     10     11     12
 13     14     15     16
```

```
STATS
```

```
Average land value in this city: 9
```

```
Maximum land value in this city: 16
```

- Run your program with `python simcity4.py` with `data_1.txt` Your program should print:

```
Sim City Land Values:
```

```
76000      0    54000    16000    83000
27000    49000    62000      0    31000
  0    48000    53000    22000    19000
```

71000	37000	63000	41000	0
83000	25000	0	16000	59000

Calculated Sim City land values:

76000	53600	54000	16000	83000
27000	49000	62000	42500	31000
46400	48000	53000	22000	19000
71000	37000	63000	41000	31400
83000	25000	36400	16000	59000

STATS

Average land value in this city: 45812

Maximum land value in this city: 83000

- Run your program with `python simcity4.py` with `data_2.txt` Your program should print:

Sim City Land Values:

94000	64000	30000	0	14000	92000
37000	49000	50000	29000	35000	0
0	88000	85000	96000	60000	22000
13000	44000	73000	0	45000	53000
20000	33000	67000	71000	82000	0
36000	0	62000	55000	44000	75000

Calculated Sim City land values:

94000	64000	30000	31600	14000	92000
37000	49000	50000	29000	35000	44600
46200	88000	85000	96000	60000	22000
13000	44000	73000	72375	45000	53000
20000	33000	67000	71000	82000	59800
36000	43600	62000	55000	44000	75000

STATS

Average land value in this city: 53227

Maximum land value in this city: 96000

- Run your program with `python simcity4.py` with `data_3.txt` Your program should print:

Sim City Land Values:

24000	57000	50000	43000
-------	-------	-------	-------

38000	0	16000	62000
51000	25000	49000	0
0	76000	19000	34000

Calculated Sim City land values:

24000	57000	50000	43000
38000	38750	16000	62000
51000	25000	49000	36000
50667	76000	19000	34000

STATS

Average land value in this city: 41839

Maximum land value in this city: 76000

## Submitting

---

Submit `simcity4.py` via eClass.

### Copyright

I. Akhmetov, J. Schaeffer, M. Morris and S. Ahmed, Department of Computing Science, Faculty of Science, University of Alberta (2023).