

Practice with 2-D Lists

```
matrix = [[1, 3, 5], [2, 4, 6], [3, 6, 9]]
```

```
matrix2 = [[5, 2, 8, 4], [-9, 0, 4, 1], [5, 6, 4, 8]]
```

1. Write a function to add up all the numbers along the upper-left to lower-right diagonal of a matrix. Assume the matrix is square.

```
def add_diagonal(grid):
```

Example. `add_diagonal(matrix)` returns 14, because $1 + 4 + 9$ is 14.

2. Write a function to multiply each odd number in the matrix by 5 (the original matrix should be altered; don't create a new matrix).

```
def mult5(grid):
```

3. Write a function to multiply all the numbers in even rows of the grid by 2 (the original matrix should be altered; don't create a new matrix).

```
def mult2even(grid):
```

4. Write a function to print the smallest number in each row of a matrix.

```
def print_smallest_in_row(grid):
```

Example: `print_smallest_in_row(matrix2)` would print 2, -9, 4.

5. Write a function to print the smallest number in each column of a matrix.

```
def print_smallest_in_col(grid):
```

Example: `print_smallest_in_col(matrix2)` would print -9, 0, 4, 1.

6. Write a function `randomize_matrix` that creates a 2x4 2-D list and fills it with exactly 2 copies of each value from 1-4. This function should return a different 2x4 matrix each time.

Hint: Randomly select the row and column to place each number into.

```
def randomize_matrix():
```

7. **Challenges:** change the print smallest/largest functions to return lists of the smallest/largest items in each row/column, rather than printing them. So problem 4 would return the list [2, -9, 4].