# CS 251 OutLab5 (Part 1) - Python Basics

## <u>Total score - 30</u>
## <u>Instructions -</u>

1. There is a file named exception.py. It contains a base exception class to be thrown in case of any error. Import the file for all the questions and raise the exception for any and all errors that might happen due to ill-formed test inputs. You are not expected to handle the exception, you are expected to just raise it! This has to be done for all the questions. You are expected to come up with all possible situations (a mark of a well-established Software Engineer) and handle them using the exception. Please note that not handling corner cases and raising the given Exception class will result in loss of marks from the autograder.
2. Running instructions for all the scripts will be provided in the template code.
3. Resources for this outlab are provided in outlab5_part_1-resources.tar.gz. Untar it using

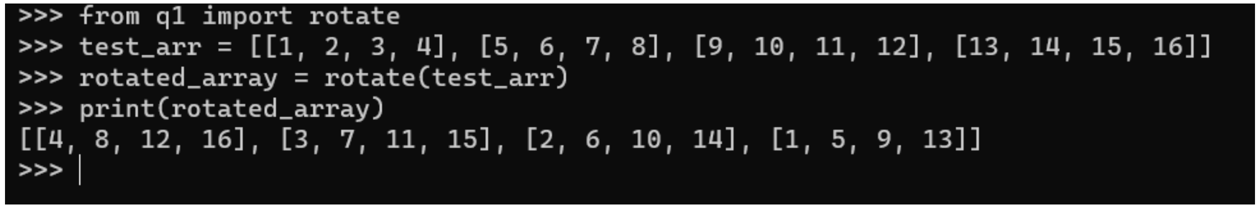
## Q.1 Rotate the matrix

A square n×n matrix (n>=1) of integers can be written in Python as a list with n elements, where each element is, in turn, a list of n integers, representing a row of the matrix. Write a function rotate(m) that takes a list representation m of a square matrix as input, and returns the matrix obtained by rotating the original matrix anti-clockwise by 90 degrees.

**Note : Don’t use Numpy library else marks will be 0.**

**Sample Input: [[1, 2, 3, 4],[5, 6, 7, 8],[9, 10, 11, 12],[13, 14, 15, 16]]**
**Output: [[4, 8, 12, 16], [3, 7, 11, 15], [2, 6, 10, 14], [1, 5, 9, 13]]**

**In the autograder, we will be importing the rotate() function alone from your q1.py script. Hence make sure that your function is able to access any declared variables or functions. To ensure this, open your python terminal and use the following commands**



```
>>> from q1 import rotate
>>> test_arr = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]
>>> rotated_array = rotate(test_arr)
>>> print(rotated_array)
[[4, 8, 12, 16], [3, 7, 11, 15], [2, 6, 10, 14], [1, 5, 9, 13]]
>>> 
```

**[ 10 Marks]**

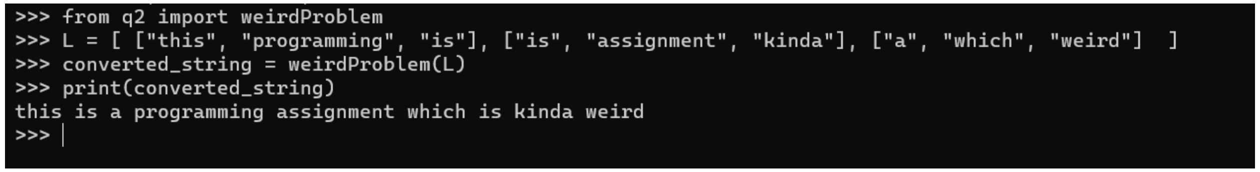
L = [ ["this", "programming", "is"], ["is", "assignment", "kinda"], ["a", "which", "weird"] ]

Function Def: weirdProblem(L)
Returns: "this is a programming assignment which is kinda weird" **without quotes.**

**Example 2:**
L = [ ["this", "programming"],["is", "assignment", "is"], ["a", "which", "kinda" ,"weird"] ]

Function Def: weirdProblem(L)
Returns: "this is a programming assignment which is kinda weird." **without quotes.**

**In the autograder, we will be importing the weirdProblem() function alone from your q2.py script. Hence make sure that your function is able to access any declared variables or functions. To ensure this, open your python terminal and use the following commands**



```
>>> from q2 import weirdProblem
>>> L = [ ["this", "programming", "is"], ["is", "assignment", "kinda"], ["a", "which", "weird"]  ]
>>> converted_string = weirdProblem(L)
>>> print(converted_string)
this is a programming assignment which is kinda weird
>>> |
```

**[ 10 Marks]**

---

**Q.3** Python supports an array of data-structures internally. One of those data-structures is a Set. It resembles the properties of a set (a well-defined collection another set such that ∀x ∊ A ⋂ B: x ∊ A and x ∊ B

Set Equality: If A and B are two sets, then A and B are said to be equal sets if and only if ∀x ∊ A: x ∊ B and ∀x ∊ B: x ∊ A.

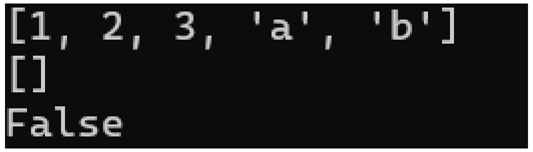
The Set data-structure in Python supports these operations by default. However, two other popular data-structures, List and Tuple don't! Your assignment (should you choose to accept it ;)) is to implement these functionality for List and Tuple too, **without using the Set data-structure**! **Any usage of functions from Set data-structure will lead to a zero**
In the accompanied folder, modify the respective functions in q3.py to accomplish these properties. We would call these functions in order to test your code. In addition to the set related functions, there is a parse_file function. It will be given a file as input. It has to read the file (format specified below) and return a list of pairs of collections (list/tuple). Refer to the function docs (within the supplied file) for more information.

The file with the test cases would be formatted as-
<list-1>    <list-2>
<list-1>    <list-2>
<tuple-1>    <tuple-2>

Each line will contain two lists/tuples (might be mixed) separated by **4 white spaces** (not tabs, for all of you Richard fans out there xD), and lines are separated by a **single newline character**. An example-

[1, 2, 3, 4]    [2, 3]
('a', 'b')    ['b', 'd']

provided in the resources.
Expected output:



```
[1, 2, 3, 'a', 'b']
[]
False
```

**[ 10 Marks]**

## Submission Instructions

**The following is what your submission directory must look like. Please follow the submission format strictly.**

The submission directory must be named
**<roll_no>_outlab5_1**

Compress it into a tarball using the following command

**tar -cvzf <roll_no>_inlab5_1.tar.gz <roll_no>_outlab5_1**

q3_test_case.txt file with your submission!!