Full Name: QUIZ 2 ANSWER KEY

Email Address (UW Net ID): @uw.edu

Section:

CSE 160 Spring 2025 - Quiz 2

Instructions:

- You have **until 4:15pm (5 minutes before the end of class)** to complete this exam.
- The exam is **closed book**, including no calculators, computers, phones, watches or other electronics.
- You are allowed a single sheet of notes for yourself.
- We also provide a syntax reference sheet.
- Turn in ***all sheets*** of this exam, together and in the same order when you are finished.
- When time has been called, you must put down your pencil and stop writing.
  - **Points will be deducted if you are still writing after time has been called.**
- You may only use parts and features of Python that have been covered in class up to this point.
- You may ask questions by raising your hand, and a TA will come over to you.

Good luck!

Question	**Topic**
Question 1	Functions
Question 2	Nested Lists
Question 3	Nested Lists

1. Given the following table with a piece of code on the left side, write out the output (what is ***printed***). If the code has an error, write `error`. If the code has a `None` output (e.g. printing a `None` type object), write `None`. If the code does not have output, write `No output`. If there are spaces in the output, mark them with an underscore (_).

Code	Output

def peanut_butter():
    return "buy peanut butter"
 	
print(peanut_butter())
	buy peanut butter

def add(a, b):
    return a + b
 	
add(2, 3)
	No output

def spotlight(name, age, major):
    print("spotlight on", name)
 	
spotlight("John")
	error

def two_strings(a, b):
    print("a=" + a)
    print("b=" + b)
 	
two_strings("b", "a")
	a=b
	b=a

2. a) Given a nested list of numbers, `data`, write code to print out the maximum number.

Example Input:	**Example Output:**
`data = [[1, 2], [7], [], [3, 5, 8]]`	8
`data = [[1, 2, 3],[4, 5, 6]]`	6
`data = [[-1],[1,2,3]]`	3

These solutions are non-exhaustive of correct answers:

Solution 1:

```
if len(data[0]) > 0:
    max_value = data[0][0]
else:
    # not great, but usually works.
    max_value = -99999999999999
for sublist in data:
    for item in sublist:
        if item > max_value:
            max_value = item
return max_value
```

Solution 2:

```
import math
# can also do float('-inf')
# (inf is not expected to be known)
max_value = -math.inf
for sublist in data:
    for item in sublist:
        if item > max_value:
            max_value = item
return max_value
```

Solution 3:

```
max_value = None
for sublist in data:
    for item in sublist:
        if max_value is None or item > max_value:
            max_value = item
return max_value
```

b) The following code is designed to calculate the total sum of all integers in the data, but it's not working correctly.

Write the current (erroneous) output of the code when `data = [[1, 2], [7], [], [3, 5, 8]]`. Then, explain two of the errors that are present.

```
total = 0
for i in data:
    for j in data:
        total = j
print(total)
```

Output: **`[3, 5, 8]`**

Error 1:

The inner loop, `for j in data`, does not loop through the inner lists. It should be `for j in i` or a similar variation to properly loop through the inner lists.

Error 2:

`total` is only set to the most recent `j` value, but instead should be incremented with either `+=` or as `total = total + j`.

3. In this problem, you will complete a function called `rotate_matrix` that takes a square matrix (nested lists) and returns a new matrix that is the result of rotating the original matrix 90 degrees clockwise**.** The solution should work for matrices of any size (e.g., 1×1, 2×2, 3×3, etc.). You can assume that the given matrix is always square (as in, it's a list with **n** sublists, each with **n** numbers).

The general concept of the function is to take advantage of the fact that rotating 90 degrees is equal to converting the columns into rows in reverse! That is:

Example Input:

```
[
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

Example Output:

```
[
    [7, 4, 1],
    [8, 5, 2],
    [9, 6, 3]
]
```

Part of the function has already been written for you. Fill in the blank lines below with the correct code to accomplish this task.

```
def rotate_matrix(matrix):
    new_matrix = []
    dim = len(matrix)
    for i in range(dim):
        new_row = []
        for j in range(dim - 1, -1, -1):
            new_row.append(matrix[j][i])
        new_matrix.append(new_row)
    return new_matrix
```