Practice Quiz 6 (Ch 11–12) Name: _____

This work complies with the JMU Honor Code. I have neither given nor received unauthorized assistance. I will not discuss the quiz contents with anyone who has not taken the quiz. Initials: _____

Question 1 (3 pts)

Consider the list of lists:

```
words = [
    ["apple", "alligator", "awesome"],
    ["bee", "barn", "berry"],
    ["cat", "celery", "candle"],
    ["dog", "dark", "dress"],
]
```

- How many rows are there? **4**
- How many columns are there? **3**
- How do you access the string `"berry"`? **words[1][2]**
- What is the output of `print(words[2])`? **['cat', 'celery', 'candle']**

Question 2 (2 pts)

Consider the dictionary of lists:

```
schedule = {
    "MWF": ["9:35am", "12:15pm", "2:30pm"],
    "TR": ["11:10am", "4:20pm"],
}
```

- List the keys, separated by commas: **"MWF", "TR"**
- Write a statement to append `"5:00pm"` to the `"TR"` list: **schedule["TR"].append("5:00pm")**

Question 3 (2 pts)

```
results = [num*2 + 1 for num in range(5)]
print(results)
```

What is output by the code? **[1, 3, 5, 7, 9]**

Question 4 (3 pts)

Consider the following function:

```
1  def timer(num):
2      print(num, end=" ")
3      if num == 0:
4          print("Done!")
5      else:
6          timer(num - 1)
7
8  timer(4)
```

- Which line number contains the base case? **3**
- How many times is the function called? **5**
- What is the output? **4 3 2 1 0 Done!**

Question 5 (3 pts)

Consider the following function:

```
1  def something(num1, num2):
2      if num2 == 1:
3          return num1
4      else:
5          return num1 + something(num1, num2 - 1)
```

- What line number contains the recursive call? **5**
- What value is returned by `something(4, 3)`? **12**
- What value is returned by `something(3, 0)`? **RecursionError**

Question 6 (5 pts)

Implement the following function. Please write clearly and indent your code using four spaces.

Hint: You can write this function in one line using a list comprehension.

```
def sum_rows(numbers):
    """Given a two-dimensional list of integers, add the integers
    in each row and return a list of the sums. If numbers is empty,
    return an empty list.

    Args:
        numbers (list): two-dimensional list of integers

    Returns:
        list: the sum of each row

    Example:
        sum_rows([[1, 34, 6], [3, 5, 8]]) returns [41, 16]
    """

    return [sum(row) for row in numbers]

    # 1 pts: build and return list
    # 2 pts: for row in numbers
    # 2 pts: append sum(row)
```

Question 7 (7 pts)

```
def skyscrapers(data, city, floors):
    """Find skyscrapers in a city that have a minimum number of floors.

    Args:
        data (list): JSON object of all known skyscrapers
        city (str): the city of interest
        floors (int): minimum number of floors

    Returns:
        list: names of skyscrapers that match the city and floors

    The JSON data is formatted as follows:
        [
            {
                "id": 12,
                "material": "steel",
                "name": "The Illinois",
                "location": {
                    "city": "Chicago",
                    "state": "Illinois",
                },
                "statistics": {
                    "height": 1609.3599853516,
                    "floors above": 528,
                    "number of purposes": 4,
                },
            },
            # other skyscraper dictionaries
        ]
    """

    return [item["name"] for item in data
            if item["location"]["city"] == city
            and item["statistics"]["floors above"] >= floors]

    # 1 pts: build and return list
    # 1 pts: for item in data
    # 2 pts: if item["location"]["city"] == city
    # 2 pts: and item["statistics"]["floors above"] >= floors
    # 1 pts: append item["name"]
```