

1. What is the output of the snippet of code below after it is executed?

```
lst = [[1, 2, 3], [4, "X", 6], [7, 8, 9]]
data = {1: "A", 2: "B", 3: "C", 4: "D", 6: "F", 7: "G", 8: "H", 9: "I"}

for row in list:
    for i in range(len(row)):
        row[i] = data[row[i]]
print(list[1][1])
```

- It will print "X"
  - It will print "B"
  - This code produces a KeyError
  - This code produces a ValueError
2. Write a function `ordered_teams` that takes the dictionary `standings` given below as input and returns a 2D list with the sublist including the team name and the team's points accumulated throughout the season. Note that each team's record in the dictionary given below is in the format [wins, regulation losses, overtime losses], and that a win is worth 2 points, an overtime loss is worth one, and a regulation loss is worth none.

- 3.
- ```
standings = {
    "Bruins": [28, 28, 8],
    "Canadiens": [30, 26, 6],
    "Penguins": [24, 30, 10],
    "Predators": [23, 32, 7],
    "Jets": [43, 16, 4],
    "Oilers": [36, 22, 4]
}
```

4. A retail store collects monthly sales data (in thousands of dollars) for three different product categories: Electronics, Clothing, and Furniture. The data is provided in a Pandas DataFrame given below.

| Months | Electronics | Clothing | Furniture |
|--------|-------------|----------|-----------|
| Jan    | 50          | 20       | 30        |
| Feb    | 55          | 22       | 28        |
| Mar    | 53          | 21       | 35        |
| Apr    | 60          | 25       | 40        |
| May    | 62          | 27       | 38        |
| Jun    | 65          | 26       | 42        |

The DataFrame above appears with the command `print(sales_df)`.

- a) Use NumPy to compute the mean sales for each product category over the six months and print the mean values.

b) Use Matplotlib to create a line plot that shows the sales for the product categories over time. (Have axis labels, a title and a legend).

5. Given the snippet of code below, find and fix the error:

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
for row in range(3):
    for col in range(3):
        print(matrix[row, col])
```

6. Write a function `countPeaks` that accepts a 2D-matrix of integers. This function should count the number of integers within the matrix that are strictly greater than their neighbors and return the total number of peaks. A neighbor is an integer to the left, right, top or bottom of the currently indexed integer.

**Sample Input to `countPeaks`:**

```
[ [10, 2, 5],
  [3, 20, 4],
  [31, 2, 14] ]
```

**Sample Output:**

5

7. What is the output of the snippet of code below?

```
matrix = [[3, 4, 6], [2, 3, 5], [2, 5, 2]]
mat = matrix
mat[0][2] = [3]
matrix[1] = [10, 5]

for row in matrix:
    for element in row:
        print(element end=" ")
print()
```

- 3 4 [3] 10 5 2 5 2
- `SyntaxError: invalid syntax`
- 3 4 6 10 5 2 5 2
- `TypeError: print() got an unexpected keyword argument`
- None of the above

8. What is the output of the following code?

```
numbers = [5, 10, 15, 20]
index = "2" + 1
print(numbers[index])
```

- This code produces a `TypeError`
- This code produces a `ValueError`
- This code produces a `NameError`
- 20

9. Trace the following snippet of code and provide the expected output.

```
tuple_a = (1, 2, [3, 4])
tuple_b = tuple_a
tuple_a[2].append(5)
tuple_b += (6, 7)
print("tuple_a:", tuple_a)
print("tuple_b:", tuple_b)
```

10. Write a function `count_digit_in_matrix(matrix, digit)` that takes a 2D list matrix of strings and an integer digit. The function should return how many times digit appears in the entire matrix.

11. Given the DataFrame `df`, filter out only the rows where Age (that appears in `df`) is greater than 23.

12. What will print after the following snippet of code is executed?

```
lst = [ [1, 2, 3, 4], [5, 6, 7, 8], [8, 10, 11, 12] ]

for i in range(1, len(lst)):
    for j in range(len(lst[i]) - 1):
        temp = lst[i][j]
        lst[i][j] = lst[i-1][j+1] * 2

        if lst[i][j] % 5 == 0 and lst[i][j] < 30:
            lst[i][j] = "multiple5"

for row in lst:
    print(row)
```

13. Consider the following snippet of code. The goal of the `count_frequencies` function is to count how many times each number appears in a list and store the frequency in a dictionary. The keys of the dictionary will be the numbers from the list, and the values will be how many times those numbers appear. However, there are **four mistakes/bugs** in the code.

- Identify the bugs in the code.
- How would you correct the code to make it return the correct frequency of each element in the dictionary?
- What will be the output of the function when the input list is `[1, 2, 2, 3, 3, 3, 4]`?

```
def count_frequencies(nums):  
    freq = []  
    for num in nums:  
        if num not in nums:  
            freq[num] += 1  
        else:  
            freq[num] = 1  
    return freq
```

14. What is the output of the following snippet of code?

```
def process_tuples(tup1, tup2):  
    result = []  
    for i in range(len(tup1)):  
        result.append(tup1[i] + tup2[i])  
    return tuple(result)  
  
t1 = (1, 2, 3)  
t2 = (4, 5, 6)  
output = process_tuples(t1, t2)  
print(output)
```

15. Consider the following snippet of code:

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
  
data = {'Student': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],  
        'Math': [85, 92, 78, 88, 95],  
        'Science': [90, 87, 82, 84, 91],  
        'English': [88, 91, 79, 85, 92]}  
  
df = pd.DataFrame(data)
```

- a) Using NumPy, compute the average score of each student across all subjects.
- b) Create a new column in the DataFrame called 'Average' that contains the average score for each student.

- c) Plot a bar graph of the average scores for all students using `Matplotlib`. The x-axis should represent the students' names and the y-axis should represent their average scores. Label the axes and give the plot a title.

16. What is the value of `d1` after the following snippet of code is executed?

```
d1= {3: 4, 5: 6, 7: 8}
for i in range(len(d1)):
    d1[i+2] = i+5
```

17. Consider a dictionary with keys as integers in increasing order (from 1) and values paired with each as a tuple of length 2. The first element of the tuple is an integer, while the second element is a list containing 2 integers. Write a function `tupleIsEqual`, which checks if the sum of the integers of the list (inside the tuple) is equal to the first element of the tuple. If not then change the contents of the list so that the sum of the integers of the list are equal to the first element of the tuple. Return a list containing all the finalized lists from smallest sum to largest.

**Example:**

Before function:

```
d1 = {1: (5, [2, 3]), 2: (4, [4, 5]), 3: (6, [1, 4])}
```

After function call:

```
d1 = {1: (5, [2, 3]), 2: (4, [4, 0]), 3: (6, [1, 5])}
return: [ [4, 0], [2, 3], [1, 5] ]
```

18. What is the output of the snippet of code below?

```
t = ((3, 2, 6), (5, 2, 1), (4, 8))
```

```
def elementsTuple(t: tuple):
    l = []
    for tup in t:
        for i in range(len(tup)):
            l.append(tup[i])
    l = tuple(sorted(l))
    return l
```

```
print(elementsTuple(t))
```

19. What is the output of the snippet of code below?

```
lst = [[5, 5, 4],
       [8, 4, 7],
       [2, 5, 1]]

def checkSum(lst):
    sum1 = 0
    sum2 = 0
    for i in range(len(lst)):
        sum1 += lst[i][i]
        sum2 += lst[i][-1-i]
    return sum1 == sum2

print(checkSum(lst))
```

20. What is the output of the following snippet of code?

```
products = {'laptop': 800, 'phone': 600, 'tablet': 400, 'headphones': 150}
discounts = {'laptop': 10, 'phone': 5, 'tablet': 20}
for product in discounts():
    if product in products:
        products[product] = products[product]*(1-discount/100)
print(products)
```

- A) {'laptop': 720, 'phone': 570, 'tablet': 320, 'headphones': 150}
- B) {'laptop': 800, 'phone': 600, 'tablet': 400, 'headphones': 150}
- C) {'laptop': 760, 'phone': 570, 'tablet': 320, 'headphones': 150}
- D) {'laptop': 720, 'phone': 570, 'tablet': 320, 'headphones': 150, 'tablet': 400}
- E) None of the above

21. What is the output of the following snippet of code?

```
tuple = (10, 20, 30 , 40)
lst = list(tuple)
lst[1] = "Hello"
print(tuple)
print(lst)
```

22. Given a dataset containing information (month, sales, expenses) about a store over a period of 6 months

- calculate the profit for each month and add it to a new column
- plot the profit for each month
- print the month in which the profit was the highest (profit, month)

23. Given a list of tuples, with each tuple representing the month, the revenue and the number of items sold; print the following:

- the sum of all the items sold in the 6 months
- the month with the highest revenue
- the average amount of items sold throughout the 6 months
- the month where revenue and items sold were closest together as well as the difference between the two values

24. You are given a 2D list (matrix) of integers of size  $m \times n$ . Write a function

`lollipop_wrap(matrix)` to return the elements of the matrix in spiral order, starting from the top-left corner and moving inward in a clockwise direction, like a lollipop.

**Example:**

Input to the function:

```
matrix = [ [1, 2, 3, 4],  
            [5, 6, 7, 8],  
            [9, 10, 11, 12] ]
```

The function returns:

```
[1, 2, 3, 4, 8, 12, 11, 10, 9, 5, 6, 7]
```

25. What is the output of the following code snippet?

```
d1 = {3: 5, 4: 5, 1:5, 11:42, 0.5:12}  
d2 = {9: 45, 16:37, 25:1, 11:51, 144:0.25}  
  
output = []  
for key in d1:  
    if d1[key]**2 in d2 and key == d2[d1[key]**2]:  
        output.append(True)  
    else:  
        output.append(False)  
  
print(output)
```

26. What is the output of the following snippet of code?

```
d = {(3,4):12, (4,5):9, (10,10):100, (2,2):4, (10,20):200}

param1 = []
for key in d:
    if key[0] * key[1] == d[key]:
        param1.append(True)
    else:
        param1.append(False)

param2 = [key[0] + key[1] == d[key] for key in d]

output = [param1[i] == param2[i] for i in range(len(param1))]

for i in range(len(output)):
    print(output[i], end=" ")
```

27. What is the output of the following snippet of code?

```
def al(n):
    return chr(65 + n)

k = {}
for i in range(26):
    k[i] = al(i)
    k[al(i)] = al(i + 1)
    if i > 0:
        k[i - 1] = al(i + 2)

print(k[12])
```

28. What is the output of the following snippet of code? What is the time complexity of the code below?

```
a = [[]]
for i in range(10):
    if i % 3 == 1:
        a.append([])
        for i in range(3):
            a[-1].append([i ** 2])
    elif i % 2 == 0:
        a[-1].append(3)

print(a[1])
```



29. What is the output of the snippet of code below?

```
count = 0
for i in range(1, 5):
    for j in range(i, 5):
        for k in range(1, j + 2):
            count += 1
print(count)
```

30. What is the output of the snippet of code below?

```
def tuples(t):
    a, b, c = t
    new_tuple = (b + c, a * 2, c - a)
    return new_tuple

original_tuple = (4, 7, 10)
result = tuples(original_tuple)
print(result)
```

31. Rewrite the snippet of code below to fix the errors within. Not all errors will be seen with the inputs provided. Think outside the box! Assume both `lst_of_keys` and `lst` will always be lists with any values. Assume `lst_of_keys` will never have repeated values. Assume `len(lst)` will be `>= len(lst_of_keys)`. Print out the result of the function.

```
lst_of_keys = [3, None, "a", False, 16.0]
lst = [[3, 5, 16], ["r", "i", "E", "c"], 145, "Computers", None,
1600.45, "yay!"]
```

```
def listToDict(lst_of_keys, lst):
    """
    Iterates over every key in lst_of_keys, assigning them to the
    value with the matching index in lst.
    """
    temp_dict = {}
    for _ in lst:
        temp_dict[lst_of_keys] == lst
    return temp_dict
```

32. What will be the output of the following code snippet? Explain why.

```
d = {'a': 10, 'b': 20, 'c': 30}
d['d'] = d.get('e', 40)
print(d)
```

33. A teacher is keeping track of students' grades using a dictionary where the keys are student names, and the values are lists of grades. The teacher wants to calculate the average grade for each student and find the student with the highest average grade. The following Python code has several bugs.

Tasks:

- Identify at least three errors in the code below.
- Explain why each error occurs
- Fix the errors

```
students = {
    'Alice': [85, 90, 78],
    'Bob': [92, 88, 95],
    'Charlie': [70, 80, 65],
    'David': [100, 98, 95]
}

highest_avg = 0
top_student = ""

for student, grades in students:
    total = 0
    for grade in grades:
        total += grade
    average = total / len(grade)

    if average > highest_avg:
        highest_avg = average
        top_student == student

print(f"The student with the highest average is {top_student} with an average of {highest_avg}.")
```

34. What is the output of the following snippet of code? Show your work for each step.

```
butterflies = {
    "name": "Monarch"
    "count": 10,
    "Location": ["forest", "lake"]
}

butterflies["count"] = butterflies.get("count", 0) + 1
if "colour" not in butterflies.keys():
    butterflies["colour"] = "orange"
butterflies.get("Location", []).append("meadow")

for key in butterflies:
    print(f"{key}: {butterflies[key]}")
```

35. There are 3 mistakes in the snippet of code given below. The expected output is 120. What are the mistakes and what are their types? Fix all mistakes.

```
def factorial(n):
    result = 1
    for i in range(1, n):
        result *= i
    return results
print("Factorial of 5 is: ", factorial[5])
```

**36.To be continued! More questions coming!!!**