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])
```

   A) It will print "X"
   B) It will print "B"
   C) This code produces a KeyError
   D) 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()
```

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

8. What is the output of the following code?

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

A) This code produces a `TypeErorr`
B) This code produces a `ValueError`
C) This code produces a `NameError`
D) 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.
    A) Identify the bugs in the code.
    B) How would you correct the code to make it return the correct frequency of each element in the dictionary?
    C) What will be the output of the function when the input list is [1, 2, 2, 3, 3, 3, 4]?

```
def count_frequences(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 `tuppleIsEqual`, 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
    A) calculate the profit for each month and add it to a new column
    B) plot the profit for each month
    C) 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:
    A) the sum of all the items sold in the 6 months
    B) the month with the highest revenue
    C) the average amount of items sold throughout the 6 months
    D) 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 x n`. Write a function `lolipop_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. Determine the output of the following snippet of code.

```
def tricky13Merge(d1: dict, d2: dict):
    temp = d1
    d1[13] = d2.get(13, 'lucky')
    value = max(d1)
    d1 = d1.update(d2)

    for key in temp:
        temp[key] = d2.get(key, temp.get(key))
    return d1, value

temp = {}
d1, d2 = {1:2, 2:3, 3:4}, {0:13, 2:13, 4:12, 13: [0, 2, 4]}
for key in d1:
    if key in d2:
        d1[key] = tricky13Merge(d2, d1)
print(d1)
print(d2)
print(temp)
```

37. Circle all possible answers that apply to each of the questions below.

Question1:
What is the output of the following code snippet?

```
total = 0
for i in range(10):
    if i % 3 == 0:
        total += i
print('3'* (total/3))
```

A) A syntax error
B) 333333
C) A Runtime error
D) A TypeError
E) None of the above

Question2:
This code snippet below creates a sorted list. Does it break at runtime?

```
matrix = [ [1, 2, 3],
           [4, 5, 6],
           [7, 8, 9] ]
_sorted_lst = [vector[v] for v in range(2) for vector in matrix]
```

A) Yes obviously
B) no
C) There is a Semantic Error
D) It crashes
E) None of the above

Question3:
Consider the matrix in Question 2. What is the output of the snippet of code below after execution?

```
num, num1 = matrix[0], matrix[1]
if not num > num1 or num1[len(num1)] < 10
     print('ok',)
```

A) It produces an IndexError
B) There is Syntax Error at the print statement only
C) There is more than one Syntax Error in the code above
D) More than one of the above
E) None of the above

38. Given an empty dictionary, write a snippet of code in two different ways to add an element to the empty dictionary.

39. Given the snippet of code below, what will the output be after it is done executing?

```
lst = [1,2,3]
letter = 'A'
num1 = [42]
num2 = 2
letter.append('B')
   print(lst + letter)
```

A) `TypeError`: can only concatenate list (not `str`) to list
B) `AttributeError`: `'str'` object has no attribute `'append'`
C) `IndentationError`: unexpected indented
D) `ValueError`: math domain error
E) All of the above
F) None of the above

40. Create a function that returns a dictionary of `n` numbers, each key should contain in a list if the number is prime or composite, even or odd, and a list of all the divisors for that number. You may create helper functions to answer this question and may use the math module and built-in python functions.

    Example:

    from the input n = 4
    The output should resemble:
    `{1: ['prime', 'odd', [1]], 2: ['prime', 'even', [1,2]], 3: ['prime', 'odd', [1,3]], 4: ['composite', 'even', [1,2,4]]}`

41. What is the output of the code below? Explain what each snippet does (each snippet is separated by a line skip)

```
def words(lst):
    dict = {}

    for word in lst:
        l_counts = {}
        for letter in word:
            if letter in l_counts:
                l_counts[letter] += 1
            else:
                l_counts[letter] = 1

        m_count = 0
        max_l = []
        for letter, count in l_counts.items():
            if count > m_count:
                m_count = count
                max_l = [letter]
            elif count == m_count:
                max_l.append(letter)


        chosen = max_l[0]
        for letter in max_l:
            if letter < chosen:
                chosen = letter

        if chosen not in dict:
            dict[chosen] = []
        dict[chosen].append(word)

    to_remove = []
    for key in dict:
        count = 0
        for word in dict[key]:
            count += 1
```

```
            if count <= 1:
                to_remove.append(key)

        for key in to_remove:
            del dict[key]

        new_keys = []
        for key in dict:
            inserted = False
            for i in range(len(new_keys)):
                if key < new_keys[i]:
                    new_keys.insert(i, key)
                    inserted = True
                    break
            if not inserted:
                new_keys.append(key)

        new_dict = {}
        for key in new_keys:
            new_dict[key] = dict[key]

        print(new_dict)

    lst = ['big', 'but', 'born', 'alt', 'any', 'little', 'lots', 'bill',
    'almost', 'giraffe', 'fox']
    words(lst)
```

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

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

43. Assuming `food_df` contains 3 columns (`apples`, `bananas` and `burgers`) with 100 rows each, what would the following code print?

```
fruits_df = food_df[[apples, bananas]]
print(fruits_df.head(10))
```

44. Write a function called `calculate_revenue` that takes three lists as input:
    - `products`: a list of product names (strings)
    - `quantities`: a list of integers representing the quantity of each product sold
    - `prices`: a list of floats representing the price per item for each product

The function should return a dictionary where the keys are product names and the values are the total revenue for each product.

Example Input:

```
products = ['Widget', 'Gadget', 'Widget', 'Tool', 'Gadget']
quantities = [3, 2, 1, 5, 3]
prices = [10.00, 15.00, 10.00, 8.00, 15.00]
```

Example Output:

```
{'Widget': 40, 'Gadget': 75, 'Tool': 40}
```

45. The following code manipulates a 2D list. Carefully trace the values of total as the code runs. What is the final output?

```
matrix = [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ]
total = 0
for i in range(len(matrix)):
     for j in range(len(matrix[i])):
          if i == j:
               total += matrix[i][j] * 2
          else: total -= matrix[i][j]
print(total)
```

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

```
def modify_tuple(t):
    x, y, z = t
    new_t = (z, x + y, y - z)
    return new_t

tuple = ((5, 2, 8), (3, 6, 1), (4, 9, 7))
lst =  []
for i in tuple:
     lst.append(modify_tuple(i))

print(lst[1][2] + lst[2][0])
```

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

```
lst = [[2, 4, 6], [1, 3, 5], [7, 9, 11]]
for i in range(len(lst)):
     for j in range(len(lst[i])):
          if i == j:
               lst[i][j] *= -1
          elif i < j:
               lst[i][j] += lst[j][i]
for row in lst:
     print(row)
```

48. Create a function called `string_slice` that takes in two values "`a`" (an integer) and "`b`" (a string). The variable "`a`" controls the maximum parts up to which the string "`b`" should be sliced, starting from `1`. The purpose of this function is to generate a 2D list with each amount of slicing subjected to the string. An example is given below.

    Example:
    `string_slice(3, "roller")` → `[["roller"], ["rol", "ler"], ["ro", "ll", "er"]]`

    You may assume that the length of the string given will be divisible by all the numbers leading up to "`a`". The function should return a 2D list. You must also refrain from using the "`.append()`" method. Once you finish defining your function, then explain and justify the time complexity of your function.

49. Create a function called `max_triangle_sum` which takes a `6x4` matrix and output the maximum sum that can be gathered from the matrix based on values in a triangular configuration. An example is given below.

    Example:

```
Matrix = [[0, 1, 0, 0, 0, 0],
          [2, 1, 3, 0, 0, 0],
          [0, 0, 0, 0, 3, 0],
          [0, 0, 0, 2, 1, 3]  ]
max_triangle_sum(matrix):
```

- `9, since 3 + 2 + 1 + 3 = 9`

`Here, 1 + 2 + 1 + 3 = 7` and hence the above configuration with sum 9 is the returned value.

You may assume that there will always be a minimal value (that <u>always</u> being 0) that will make it easier to distinguish occurrences of triangles. You may also assume that triangles will not be merged into one another and that the triangles will never be upside down. Note that there may be more than just 2 triangles within the matrix. You must use nested loops to create this function, and <u>you may **not** use the max function</u>.

Once you are done, please give the time complexity of your function.

50. You want to cook a matcha cheesecake for your mom. However, the cooking book has all of the steps completely reversed and numbered +2. Fix the cookbook (which is actually a dictionary where the keys are the steps and the values are the step numbers) so you can cook the cheesecake! You also want to print a list with the steps for better readability. Dictionaries confused you...

    Hurry! Your mom is coming soon!

51. You have a csv file called `car_velocity`, which includes data of a car's velocity over time. You are to:
    - Read the `csv` file
    - Drop missing `NaN` values
    - Extract `time` and `velocity` columns
    - Compute summary statistics like (`min, max, mean, count`)
    - Plot a scatter plot with the labeled axes

52. In the code below, you have a 2-dimensional list that will always be of size `n*n`. You are to find 2 bugs in the code, and after fixing them, find out what prints.

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

num = []
for i in range(len(num)):
    for j in range(len(num)):
        if num[i][j] % 2 = 0:
            num.append(num[i][j])

print("nums are:", nums)
```

53. Given the snippet of code below determine the output. Justify your reasoning. Write out your though process in a systematic manner. Hint: the name of the function.

```
lst = [   [1, 0, 1],
          [4, 3, 9],
          [2, 7, 1] ]

def tuplePositionPair(lst):
    result = []
    for i in range(len(lst)):
        row = []
        for j in range(len(lst[i])):
            row.append((lst[i][j], i - j//2))
        result.append(row)
    return result
```

54. Say you have a CSV file named "random-data.csv". Go through the beginning process of handling such data which we have seen in class. For example, perform the importing, reading, describing, head and tail, & displaying. Also perform a very simple cleaning of the file (NaN values dropped). Display the dataframe after every step when "deemed" necessary. Assume that printing is not necessary.

55. Using `NumPy`, calculate the average temperature in Fahrenheit. To make Celsius temperature, Fahrenheit use the following formula: F = (C * (9/5)) + 32).
Once complete, create a line graph with `x = Day of the week` and `y = temp in F`

`Data =`

Day	**Temperature (C)**
Monday	22
Tuesday	19
Wednesday	25
Thursday	18
Friday	21
Saturday	23
Sunday	20

create a dictionary called `data` with the table above (keys are the days, values are the temperature)
and then call the command below to create DataFrame `df`.
`df = pd.DataFrame(data)`

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

```
def update_inventory(item, quantity):
     inventory[item] = quantity
inventory = {'apple': 10, 'banana': 5}
update_inventory('orange' 3)
print(inventory)
```

A) {‘apple’: 10, ‘banana’: 5, ‘orange’: 3}
B) Syntax Error
C) Runtime Error
D) Semantic Error
E) None of the above

57. Write a program that takes an integer `n` as input from the user and prints a hollow diamond pattern with `2n-1` rows using asterisk `(*)`. Here is an example of what the diamond should look like for input `n=5`.

```
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

58. Given the following code below, what is the output after execution? Select all that apply.

```
def calculate_rectangle_properties():
    length = input("Enter the length of the rectangle: ")
    width = input("Enter the width of the rectangle: ")

    area = length * width
    perimeter = 2 * (length + width)

    print(f"The area of the rectangle is {area}.")
    print(f"The perimeter of the rectangle is {perimeter}.")

calculate_rectangle_properties()
```

A) Name Error
B) Type Error and Runtime Error
C) Parse Error and Syntax Error
D) Syntax Error
E) None of the above

59. Given a list of dictionaries containing the information of different destinations of flights, use `pandas` to turn the data into a DataFrame (i.e. `pd.DataFrame(lst_o_dict)`) after making a function `flights_data` that takes in the list of dictionaries `lst_o_dict` to compute their total number of flights per destination. Return the results as a nested dictionary.

Example:
input:

```
lst_o_dict = [ {'destination': 'USA', 'flights': 11},
          {'destination': 'France', 'flights': 19},
          {'destination': 'USA', 'flights': 20},
          {'destination': 'France', 'flights': 10},
          {'destination': 'Morocco', 'flights': 27},
          {'destination': 'China', 'flights': 35},
          {'destination': 'Morocco', 'flights': 9} ]
```

output:

```
{'total_flights_per_destination': [{'destination': 'China', 'flights': 35},
{'destination': 'France', 'flights': 29}, {'destination': 'Morocco',
'flights': 36}, {'destination': 'USA', 'flights': 31}]}
```

60. Write a function `dict_opp` that combines 3 different dictionaries `d1`, `d2`, `d3` into a new dictionary `new_dict`.
    A) if the type of the key is an integer, the value of the key will be the sum of everything in the lists.
    B) if the type of the key is a string, it should not appear in the new list

    Note that the keys should be in all of the dictionaries. You may assume there are only integers in the lists

    input example:
```
d1 = {"a": [68, 67, 45], 1: [87, 78, 56], 2: [98], 66: [66, 67, 0, 3]}
d2 = {"a": [21, 25, 36], 1: [3, 4, 6], 2: [45], 66: [3]}
d3 = {"a": [7, 9, 8], 1: [45, 67], 3: [3], 66: [2, 6, 8]}
```

    output:
```
{1: 346, 66: 155}
```

61. What is the output of the snippet of code below after execution?

```
def transform_tuple(data):
    a, (b, c) = data
    b = list(b)

    for i in range(len(b)):
        b[i] += c[i]

    new_tuple = (sum(b) * a, tuple(b), c[::-1])

    return new_tuple

data = (3, ((4, 6, 8), (5, 7, 9)))
result = transform_tuple(data)

print(data)
print(result)
```

62. What is the output of the code below after execution?

```
def nested_loop_modify(n):
    lst = []

    for i in range(1, (n + 1)):
        amount = []
        for j in range(n):
            amount.append(i * j)
        lst.append(amount)

    for x in range(n):
        for y in range(x, n):
            lst[y][x] += (x + y)

    return lst

result = nested_loop_modify(4)
for amount in result:
    print(amount)
```

63. Given a CSV file containing daily temperature readings for a city over a year
    A) Load the data using `pandas` and calculate the average monthly temperature using `NumPy`.
    B) Create a `Matplotlib` line plot showing the monthly average temperatures.

64. You are given a dictionary with a student's subjects as keys and grades as values. Here is an example:

```
Alice = {"Math": 28, "Science": 48, "English": 60, "French": 53, "Humanities": 57}
```

You are also given a dictionary with the classes eligible for extra credit. Here is an example:

```
Credit = {"Math": True, "Science": False, "English": False, "French": True,
"Humanities": True}
```

If they are eligible for extra credit, 15% of the grade is added to the total of the corresponding subject.

- Calculate the final grade based off of their credit eligibility and update the value in the student's dictionary
- To pass the class they need over 60%
- Print the subjects that the student passed with their final grade using the format: `{subject}: {grade}% — PASSED`

65. Given the code below, determine the final value of `result` after all loops finish. Show the iteration steps in your answer.

```
result = 0
for i in range(1, 4):
    for j in range(i, 4):
        for k in range(1, j+1):
            result += (i * j - k)
```

66. Write a Python function named `wordCheck` with two inputs: `grid` and `word`, that searches for the given word in a `3x3` 2D character grid. The word can be found horizontally or vertically.

Example input:

```
grid = [  ['c', 'a', 't'],
          ['d', 'o', 'g'],
          ['r', 'a', 't'] ]
word = "cat"
```

Output:

```
True (found in row 0)
```

67.

A) Create a 5x5 2D list using list comprehensions, where each element is initialized to -1.
B) What is the time complexity of your code in part a).

68. Write a function called `group_reversed` that takes a list of words and groups them into a dictionary where the keys are sorted versions of the words, and the values are the inverted version of the words IF they exist in the list. **Hint:** you can reverse a string by doing str[::-1].

Input:

```
group_anagrams (["listen", "netsil", "rat", "boar", "rta"])
```

Output:

```
{"listen": "netsil", "rat": "tar"}
```

69. What is the output of the following snippet of code after execution?

```
def modify_tuple(t):
    t[1][0] += 5
    return (t[0] * 2, t[1])

t1 = (3, [4, 5])
t2 = modify_tuple(t1)

print(t1)
print(t2)
```

A) (3, [4, 5]) and (6, [9, 5])
B) B) (3, [9, 5]) and (6, [9, 5])
C) C) (3, [4, 5]) and (6, [4, 5])
D) D) (6, [9, 5]) and (6, [9, 5])

70. The function below should group a list of strings in a dictionary by length where the keys are integers representing the length of the word and the values are lists containing all words with the given length. The function should return the said dictionary. You can only change the given lines of code.

Example Input:

```
> ["apple", "banana", "pear", "grape", "kiwi", "peach"]
```

Example Output:

```
> {5: ['apple', 'grape', 'peach'], 6: ['banana'], 4: ['pear',
'kiwi']}
```

NOTE: This function DOES compile. It contains 2 Semantic Errors. Fix the two errors.

```
def group_words_by_length(words):
    grouped = {}
    for word in words:
        length = len(word)
        if length in grouped:
            grouped[length] += word
        else:
            grouped[length] = word

    return grouped

print(group_words_by_length(["apple", "banana", "pear", "grape",
"kiwi", "peach"]))
```

71. Create a function that takes a list of strings and removes all asterisks (*) by modifying the original list. The function modifies the list in place, thus it does not return anything. Do not use the `replace()` built-in python function for this exercise.

Example input:
```
> ['wo*rd','sal*m*on','mat*h**']
```

Example output:
```
> ['word','salmon','math']
```

72. Create a coordinate system that asks for an input `n` from the user for an `n x n` grid and return the coordinates in 2D list format, starting from `(0, 0)` at the bottom left of the grid. The coordinates should increment the `x` values from left to right and move upward along the `y`-axis. The coordinate system will start at 0 on both axes. If the input is an odd number, replace the tuple in the middle of it with `" ** "`. (There is 1 space on either side.)

Input: `3`
Ouput:
```
[(0, 2), (1, 2), (2, 2)]
[(0, 1),   **  , (2, 1)]
[(0, 0), (1, 0), (2, 0)]
```

73. What is the output of the following snippet of code below?

```
def someFunction(n):
     lst = []
     for i in range(n):
          row = [j + 1 + i * n for j in range(n)]
          lst.append(row)

     for i in range(len(lst)):
          for j in range(len(lst[i])):
               for k in range(2, int(lst[i][j]**0.5) + 1):
                    if lst[i][j] % k == 0:
                         lst[i][j] = 0
     return lst

modified = someFunction(5)
for row in modified:
     print(row)
```

74. Given an upper integer limit of limit, write a function that returns all Pythagorean triples (integers that respect a^2 + b^2 = c^2) that respect this limit. The time complexity of your function must be within O(n^2) operations.

75. Given the snippet of code below, determine the number of mistakes in the code.

```
word = []
def asList(string):
    for i in range(0, len(string)):
    word.append(string[i])

asList(input'Enter any word')
```

A) 1
B) 2
C) 3
D) 4

76. The given code below does not work as intended. Take the code and provide a corrected version that works properly. The code is made to take in tuples and add them to a list of items. This list is then read and the tuples are printed with the highest value in its last position.

```
import random
tuples = []
words = ['this', 'code', 'sucks']

for i in range(2):
    tuples.append((random.randint(1, 10), words[i] ))

def highestTuple(tuples):
    tuples.append(tuple(((input('Give a number')),\
      (input('now give a word associated to that word')))))
    tuples.sort
    print(tuples[-1])

highestTuple(tuples)
```

77. Find 3 errors in the code below, explain them, and label them as `'Syntax'`, `'Runtime'` or `'Semantic'` errors. The purpose of the functions is to add each of the elements together and output the sum.

```
def mySum1(m):
    output = 0
    for x in m:
        output += float(x)
    return x

def mySum2(m):
    output = 0
    length_m = len(m)
    return sum([[int(x)] for x in range(length_m)])

def mySum3(m):
    output = 0
    for x in m[-1::-1]:
        if type(x) == list:
            output += x[-1]
        else:
            output += float(x)
    return output

def mySum4(m):
    output = 0
    for x in m:
        if type(x) == list:
            for k in x:
                output += k
        else:
            output += float(x)
        return output

m = [1,'2.5',[10],-1]

mySum1(m)
mySum2(m)
mySum3(m)
mySum4(m)
```

78. Write a function called `adjacents` that takes in the parameter `matrix` (a `nxn` 2D list of `ints`). All values in the matrix have one digit except for 1, this value has 2 digits. The function: `adjacents(matrix)` should return the number of adjacent tiles to the 2 digit number (horizontal, vertical and diagonal). The function must have a runtime of O(n^2) or less.

Sample inputs:

```
matrix = [[0,0 ,0],          → adjacents(matrix) should return 8
          [1,10,0],
          [0,0 ,1]]

matrix = [[1,2,0 ],          → adjacents(matrix) should return 3
          [0,0,0 ],
          [0,0,12]]
```

79. You are given a 2D list where each inner list contains a student's name, their class, and their scores across various subjects. Write a function `organize_students` that takes this list and returns a dictionary where:
    A) The keys are the class names.
    B) The values are dictionaries that contain:
        - The keys as student names.
        - The values as the average score of the student.

Example input:

```
students = [["Alice", "Math", 80, 90, 70],

           ["Bob", "Math", 85, 92, 88],

           ["Charlie", "Science", 95,100, 90],

           ["David", "Science", 60, 75, 80]]
```

Expected Output:

```
{"Math": {"Alice": 80.0,"Bob": 88.33}, "Science": {"Charlie": 95.0,
"David": 71.67}}
```

80. Write a function that returns True if a given number exists in a 2D list and False otherwise. Here is the function header: `def exists(matrix, target):`

81. You are provided with a CSV file (`people.csv`) that contains data about people, including their gender and age. Your goal is to:
    1. Read the CSV file into a Pandas DataFrame.
    2. Filter out only the rows where the gender is "Female"
    3. Further filter the DataFrame to include only females who are older than 20 years.
    4. After filtering it all, display 10 random values out of these older than 20 females

82. Consider the following Python code snippet:

```
my_tuple = ([1, 2, 3], [4, 5, 6], [7, 8, 9])
my_tuple[1] = [10, 11, 12]
my_tuple[1].append(1)
my_tuple[0][1] = 20 print(my_tuple)
```

    What is the error message, and how can it be fixed?

83. You have al ist of tuples and each tuple has a food and a price int. For example: (apple, 1.25). Prompt the user to enter how many of each they bought. Then print a dictionary, where the keys as the food and the values as the total price. You may assume that the list of tuples for food is already given (i.e you can hardcode it in).

    Input Specification:

    Input the food and their number on a single line separated by space. Once, done buying the food, input the word DONE.

    Example Input:

```
food_lst=[("apples",1.50),("milk",4.50),("orange",1.25),("chicken",8.50)] # hardcoded
```

```
apples 3
milk 4
orange 1
chicken 2
orang 2
DONE
```

    Output:

```
{"milk": 18, "chicken": 17, "orange": 3.75, "apples": 4.5}
```

84. Given the snippet of code below, what is the output after execution?

```
d = {"a": "apple", "b": "banana", "c": "cherry"}

def manipDict(d):
    result = {}
    for key in d:
        value = d[key]
        if len(value) % 2 == 0:
            if "b" in value:
                result[key] = value[::2]
            else:
                result[key] = value[::-1]
        else:
            if "a" in (value):
                result[key] = value[1:] + value[0]
            else:
                result[key] = value.upper()
    return result

print(manipDict(d))
```

A) {'a': 'pplea', 'b': 'bnn', 'c': 'yrrehc'}
B) {'a': 'apple', 'b': 'banana', 'c': 'cherry'}
C) {'a': 'elppa', 'b': 'bnn', 'c': 'CHERRY'}
D) {'a': 'apple', 'b': 'bnana', 'c': 'CHERRY'}
E) None of the above

85. What is the output for the following code?

```
lst = [(num, [num**2, num+1]) for num in range(1,3)]
num = lst[1][1]
num[0] += 7
print(num,lst)
```

86. Given DataFrame `df` with columns `"height"`, `"age"`, `"weight"`, and `"eye colour"`, plot using `matplotlib` a scatter plot with `age` as your `x` and `weight` as your `y` with red coloured dots.