Lab 3.Function (Questions)

Assignment Submission Guidelines

Steps to Submit Your Lab Assignments Using Google Colab

1. **Submission Deadline:**
    - All assignments must be submitted **no later than 23:59PM tonight**.
    - Late submissions will not be accepted unless prior arrangements have been made by the TAs.
2. **Complete Your Work:**
    - Finish your work in Google Colab and ensure that your notebook is saved.
3. **Generate a Shareable Link:**
    - Once your work is ready for submission, click on the **"Share"** button in the upper-right corner of your Colab notebook.
    - Under **General access**, change the setting to **"Anyone on the internet with the link"** and select **"Editor"** as the permission level.
    - Copy the generated shareable link.
4. **Submit Your Assignment on Canvas:**
    - Log in to **Canvas** and navigate to the assignment submission page.
    - Paste the copied link into the submission field on Canvas and submit your assignment.
5. **File Naming Convention:**
    - Ensure that you naming is as follows: `Lastname_Firstname_AssignmentName`
    - Example: `Alex_John_Lab3.ipynb` and `Alex_John_Lab3.pdf`
6. **Technical Issues:**
    - If you encounter any technical issues with Canvas or your submission, contact the TAs immediately **before the deadline** to avoid penalties.

Questions

Question 1 (5 Points)

Write a function named square that takes a number as input and returns the square of that number.

In [160…

```
# The function, using type annotation.
def square(num:int):
    return num**2 # Return the square of the inpue.

print(square(7))
```

```
49
```

Question 2 (5 Points)

Write a function named power that takes two numbers, base and exponent, as input and returns the result of raising base to the power of exponent.

In [161…

```
# Defining the function.
def power(power_base:int, power_exp:int):
    return power_base**power_exp # Returning the calculation.

print(power(2,10))
```

```
1024
```

Question 3

Write a function named find_max that takes a list of numbers as input and returns the maximum value in the list.

In [162…

```
# Defining the function.
def max_num(*args):
    # Returning a list and the max value just to make the print statement pr
    return list(args), max(args)

# Assigning a variable for the example.
call_function = max_num(1,5,2,81,4)

# A fancier demonstration of the function that includes a print of all the r
print(f"The max of the numbers: {call_function[0]} is {call_function[1]}.")
```

```
The max of the numbers: [1, 5, 2, 81, 4] is 81.
```

Question 4 (5 Points)

Write a function named count_vowels that takes a string as input and returns the count of vowels (a, e, i, o, u) in the string.

In [163…

```
# The list of vowels we will check against.
vowels = ['a', 'e', 'i', 'o', 'u']

# The vowel counting function.
def count_vowels(input_string: str):
    # Vowel count starts at zero.
    vowel_count = 0
    # We will iterate over the entire string.
    for letter in input_string:
        # We check if each letter (in lower case) is in the vowel list.
        if letter.lower() in vowels:
            # Adding one to the vowel count.
            vowel_count += 1
    return vowel_count

test_string = 'BRUH MOMENT'

print(f'There are {count_vowels(test_string)} vowel(s) in {test_string}')
```

```
There are 3 vowel(s) in BRUH MOMENT
```

Question 5 (10 Points)

Write a function called `calculate_average` that takes a list of numbers as a parameter and returns the average of those numbers.

In [164…

```
# Defining the function.
def calculate_average(*args):
    # Using built in functions for the calculation.
    average = sum(args)/len(args)
    # Returning the average.
    return average

print(calculate_average(1,10,100,1000))
```

```
277.75
```

Question 6 (10 Points)

Write a function called `calculate_area` that takes the radius of a circle as a parameter and returns the area of the circle. Assume that the value of pi is 3.14.

In [165…

```
# Defining the function.
def calculate_area(radius: int):
    # Pi approximation.
    pi_approx = 3.14159
    # Returning the calculation.
    return pi_approx*radius**2

calculate_area(1)
```

Out[165…

```
3.14159
```

Question 7 (10 Points)

You are working on an application for a movie theater. The application needs a feature where users can specify preferences for their movie experience.

Write a Python function named movie_preferences that:

- Accepts an arbitrary number of keyword arguments.
- Processes these keyword arguments to set preferences for the movie-goer.
- Returns a dictionary with the preferences.

For example, if the function is called with the arguments seat="Front Row", snack="Popcorn", and drink="Soda", the function should return the dictionary:

In [166…

```
{
    'seat': 'Front Row',
    'snack': 'Popcorn',
    'drink': 'Soda'
}
```

Out[166…

```
{'seat': 'Front Row', 'snack': 'Popcorn', 'drink': 'Soda'}
```

Write your solution for the movie_preferences function.

In [167…

```
def movie_preferences(**kwargs):
    return dict(kwargs)

movie_preferences(genre="Comedy", spiderman="Toby", food="No", theater='AMC'
```

Out[167…

```
{'genre': 'Comedy', 'spiderman': 'Toby', 'food': 'No', 'theater': 'AMC'}
```

Question 8 (10 Points)

You've been hired as a software developer for "MusicFest," a company that organizes music concerts. They want a system where they can plan concerts and list the details. You need to create a Python function named organize_concert that captures and displays the concert details.

The function should:

1. Accept a mandatory positional argument date which specifies the date of the concert.
2. Accept an arbitrary number of artists/bands that will perform at the concert.
3. Accept keyword arguments for other details of the concert such as genre, duration, and venue.
4. Print the concert details in a structured format.

For instance, if the function is called with the arguments `15th October`, `Coldplay`, `Ed Sheeran`, genre=`Rock/Pop`, duration=5, and venue="Stadium", it should print:

In [168…

```
Organizing a Rock/Pop concert on 15th October for 5 hours.
Venue: Stadium
Artists include:
- Coldplay
- Ed Sheeran
```

```
  Cell In[168], line 1
    Organizing a Rock/Pop concert on 15th October for 5 hours.
               ^
SyntaxError: invalid syntax
```

Write the organize_concert function based on the given requirements.

In [169…

```
# Just noting that Christopher advised kwargs are not needed in this example

def organize_concert(date: str, genre: str, duration:str, venue:str, *args):
    # The long string.
    printout_string = f'''
We are organizing a {genre} concert on {date}!
The show will take place at {venue}.
Artists include:'''
    print(printout_string)
    # This for loop iterates over the artists from args.
    for artist in args:
        print('-',artist)
    return None

organize_concert('October 14th', 'Techno', '32 Hours','Arlington Municipal C
                 'Justin Bieber', 'LöC', 'The Wiggles', 'Spongebob Squarepar
```

```
We are organizing a Techno concert on October 14th!
The show will take place at Arlington Municipal Court.
Artists include:
- Justin Bieber
- LöC
- The Wiggles
- Spongebob Squarepants
```

Question 9 (10 Points)

You've been hired by a company called "Fruitful Endeavors" to write a function for their new "Talkative Fruit Scale" product. This scale not only weighs fruits but also comments on them!

- **Write a Python function named talkative_fruit_scale that:**

1. Accepts two arguments: the fruit_name and its weight in grams.
2. Returns a string that comments humorously on the fruit based on its weight.

Here are the weight guidelines:

- Less than 100 grams: "That's a lightweight [fruit_name]!"
- 100 to 300 grams: "That's a decently-sized [fruit_name]!"
- More than 300 grams: "Whoa! That's a [fruit_name] heavyweight champion!"

For instance, if the function is called with "apple" and a weight of 50, it should return: "That's a lightweight apple!"

In [170…

```python
# The function.
def talkative_frut_scale(fruit: str, weight: int):
    # The biggest fruit.
    if weight > 300:
        return f"Whoa! That {fruit} is a heavyweight champion!"
    # The inbetween fruit.
    elif weight >= 100 and weight <= 300:
        return f"That's a decently sized {fruit}"
    # Everything else is small fruit.
    else:
        return f"That's a light weight {fruit}"

talkative_frut_scale('apple', 3002)
```

Out[170…

```
'Whoa! That apple is a heavyweight champion!'
```

Question 10 (10 Points)

You're working for a company called **Magic Weather Ball**. This ball doesn't just predict the weather; it gives quirky advice based on the forecast!

Write a Python function named weather_advice that:

1. Accepts one argument: forecast, which can be "sunny", "rainy", or "cloudy".
2. Returns a humorous piece of advice based on the forecast. Guidelines:

- "sunny": "Wear sunglasses, and maybe some sunscreen. Remember, you're not a tomato!"
- "rainy": "Take an umbrella, or you'll be singing in the rain... without the singing!"

- "cloudy": "It might rain, or it might not. Life's full of surprises, isn't it?

In [171…

```
def weather_advice(forecast: str):
    # Evaluating the conditions.
    if forecast.lower() == 'sunny':
        return "Wear sunglasses, and maybe some sunscreen. Remember, you're
    elif forecast.lower() == 'rainy':
        return "Take an umbrella, or you'll be singing in the rain... withou
    elif forecast.lower() == 'cloudy':
        return "It might rain, or it might not. Life's full of surprises, is
    # Invalid condition.
    else:
        return 'invalid forecase, try again.'

weather_advice('sunny')
```

Out[171…

```
"Wear sunglasses, and maybe some sunscreen. Remember, you're not a tomato!"
```

Question 11 (10 Points)

Write a Python function, **HoursMinutes_to_Seconds**, that converts given hours and minutes into total seconds

In [172…

```
# Defining the function.
def HoursMinutes_to_Seconds(hours, minutes):
    # Convert hours to seconds.
    seconds_from_hours = hours * 3600
    # Convert minutes to seconds.
    seconds_from_minutes = minutes * 60
    # Add them together.
    total_seconds = seconds_from_hours + seconds_from_minutes
    return total_seconds

HoursMinutes_to_Seconds(5,2)
```

Out[172…

```
18120
```

Question 12 (10 Points)

Write a Python function to check if a given number is prime or not. The function should return True if the number is prime, otherwise False.

In [173…

```
# The prime function.
def is_prime(n):
    # If the number is 1 or less, it can't be prime.
    if n <= 1:
        return False

    # Try dividing n by every whole number starting at 2.
    # If ANY division comes out even (no remainder), then n is not prime.
    test_number = 2
    while test_number < n:
        if n % test_number == 0:
            return False  # Found a divisor, so it's not prime.
        test_number = test_number + 1

    # If we never found a divisor, then n is prime.
    return True
```