# Viva Questions

**Lab 1a: Write a python program to find the best of two test average marks out of three test's marks accepted from the user.**

1. How does your program accept user input for the three test marks?
2. What steps does your program take to calculate the average marks for each test?
3. How does your program determine the best of the two test averages?
4. Could you explain the use of any conditional statements or loops in your program?
5. How would your program handle invalid inputs, such as non-numeric characters or negative values for test scores?

**Lab 1b: Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.**

1. How does your program determine whether the given number is a palindrome or not?
2. Could you explain the approach your program uses to count the occurrences of each digit in the input number?
3. How does your program handle potential errors or invalid inputs, such as non-numeric characters or negative values?
4. Does your program utilize any data structures or specific techniques to count the occurrences of digits efficiently?
5. What optimizations, if any, have you implemented to enhance the efficiency of your program, especially in counting the occurrences of digits?

**Lab 2a: Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed**.

1. Could you explain the concept behind the function F, where Fn = Fn-1 + Fn-2?
2. How does your Python program verify and handle the condition for the input value (N) to ensure it's greater than zero?
3. Could you explain the process within your program that passes the user-provided value (N) to the defined function F?
4. Does your program utilize any iterative or recursive methods to calculate the function Fn = Fn-1 + Fn-2?
5. How does your program ensure the display of an appropriate error message when the input value does not satisfy the condition N > 0?

**Lab 2b**: **Develop a python program to convert binary to decimal, octal to hexadecimal using functions.**

1. Can you explain the specific functions you've created for converting binary to decimal and octal to hexadecimal?
2. How does your program validate the input received from the user to ensure it represents a valid binary or octal number before performing the conversion?
3. Could you describe the algorithms or mathematical operations used within the functions to perform the conversions accurately?
4. How does your program manage potential errors or unexpected inputs during the conversion process?
5. Do your functions handle large binary numbers or octal values effectively, or are there limitations to the size of the numbers they can process?

**Lab 3a**: **Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters**

1. How does your program determine the number of words in the provided sentence?
2. Could you explain the process your program follows to identify and count the number of digits within the sentence?
3. What approach does your program take to calculate the count of uppercase letters in the sentence?
4. How does your program differentiate between lowercase and uppercase letters while counting their occurrences?
5. Does your program account for potential edge cases or unexpected inputs, such as special characters or punctuation within the sentence?

**Lab 3b**: **Write a Python program to find the string similarity between two given strings**

1. What specific metric or approach does your program use to calculate the similarity between two strings?
2. Can you explain how your program handles variations in string lengths when computing their similarity?
3. Could you discuss the types of string similarity measurements or algorithms your program is utilizing?
4. How does your program account for cases where the strings have similar content but different structures or order of words?
5. Does your program consider the impact of case sensitivity or different character types (such as Unicode characters) in determining string similarity?

### Lab 4a: Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib.

1. How does your program utilize Matplotlib to create a bar plot in Python?
2. Could you explain the data representation used in your program for creating the bar plot?
3. What customization options or attributes does your program employ to enhance the visualization of the bar plot (e.g., colors, labels, axes formatting)?
4. Does your program allow for multiple bars, grouped bars, or stacked bars in the plot? If so, how does it achieve this functionality?
5. How does your program handle potential errors or unexpected data types while creating the bar plot?

### Lab 4b: Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib.

1. How does your program use Matplotlib to create a scatter plot in Python?
2. What type of data structure does your program utilize for generating the scatter plot?
3. Could you explain the customization options applied in your program to enhance the visual aspects of the scatter plot (e.g., marker styles, colors, labels, axis formatting)?
4. Does your program handle additional dimensions in the scatter plot, such as size or color of markers to represent another variable? If so, how is this accomplished?
5. How does your program address potential issues or unexpected data types when creating the scatter plot?

### Lab 5a: Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib.

1. How does your program utilize Matplotlib to create a histogram plot in Python?
2. What type of data representation or structure does your program use to create the histogram?
3. Could you explain the customization options implemented in your program to enhance the visual aspects of the histogram (e.g., bins, colors, labels, axes formatting)?
4. Does your program provide flexibility in setting the number of bins or handling different bin sizes in the histogram plot? If so, how is this managed?
5. How does your program manage potential errors or unexpected data types while creating the histogram plot?

### Lab 5b: Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib.

1. How does your program utilize Matplotlib to create a pie chart in Python?

2. What type of data representation does your program employ for generating the pie chart?
3. Could you explain the customization options your program uses to enhance the appearance of the pie chart (e.g., colors, labels, explode, shadow effects)?
4. Does your program handle the labeling of slices in the pie chart, especially when there are many slices or small portions? If so, how is this managed?
5. How does your program address potential errors or unexpected data types while creating the pie chart?

### Lab 6a: Write a Python program to illustrate Linear Plotting using Matplotlib.

1. How does your program use Matplotlib to create a linear plot in Python?
2. Could you explain the data representation and format used within your program to illustrate the linear plot?
3. What customization options or attributes does your program utilize to enhance the appearance of the linear plot (e.g., line styles, colors, labels, axes formatting)?
4. Does your program handle multiple lines on the same plot or incorporate markers along the plotted line? If so, how is this implemented?
5. How does your program manage potential errors or unexpected data types while creating the linear plot?

### Lab 6b: Write a Python program to illustrate liner plotting with line formatting using Matplotlib

1. How does your program use Matplotlib to perform linear plotting with line formatting in Python?
2. What line formatting options or attributes does your program employ to enhance the appearance of the plotted lines (e.g., line styles, colors, thickness)?
3. Could you explain how your program differentiates and illustrates various lines using different formatting in the same plot?
4. Does your program allow for the inclusion of markers or points along the plotted lines, and how are these incorporated with the line formatting?
5. Does your program allow for the inclusion of markers or points along the plotted lines, and how are these incorporated with the line formatting?

### Lab 7: Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions

1. How does your program utilize Seaborn's aesthetic functions to customize plots in Python?

2. Could you explain the types of customizations applied using Seaborn's aesthetic functions (e.g., changing color palettes, modifying plot styles, adjusting plot elements)?
3. What are the advantages of using Seaborn's aesthetic functions over standard plotting libraries for plot customization?
4. Does your program demonstrate the application of aesthetic functions in multiple types of plots (e.g., bar plots, line plots, scatter plots)? If so, how is this achieved?
5. How does your program handle potential errors or unexpected inputs when using Seaborn's aesthetic functions for plot customization?

### Lab 8: Write a Python program for plotting different types of plots using Bokeh

1. How does your program utilize Bokeh to create different types of plots in Python?
2. Which types of plots are demonstrated in your program using Bokeh?
3. Could you explain the approach your program uses to handle and format the various plot types (e.g., line plots, scatter plots, bar plots) in Bokeh?
4. Does your program showcase interactivity features or tools available in Bokeh for the created plots? If so, how are these features implemented?
5. How does your program handle potential errors or unexpected data inputs when creating different types of plots using Bokeh?

### Lab 9: Write a Python program to draw 3D Plots using Plotly Libraries

1. How does your program use the Plotly library to create 3D plots in Python?
2. Which types of 3D plots are demonstrated in your program using Plotly (e.g., 3D surface plots, scatter plots, contour plots)?
3. Could you explain the approach your program takes to handle and format the different types of 3D plots in Plotly?
4. Does your program showcase any interactive or customizable features available in Plotly for the created 3D plots? If so, how are these features implemented?
5. How does your program handle potential errors or unexpected data inputs when creating various 3D plots using Plotly?

### Lab 10a: Write a Python program to draw Time Series using Plotly Libraries.

1. How does your program utilize Plotly to create Time Series plots in Python?
2. Which specific features of Plotly does your program use to handle time-based data in creating Time Series plots?
3. Could you explain the process your program follows to format and represent Time Series data within the Plotly environment?
4. Does your program offer any interactive elements for the Time Series plots, such as zooming, panning, or tooltips? If so, how are these implemented?

5. How does your program manage potential errors or unexpected data formats when creating Time Series plots using Plotly?

### Lab 10b: Write a Python program for creating Maps using Plotly Libraries

1. How does your program utilize Plotly libraries to create maps in Python?
2. Which types of maps are demonstrated in your program using Plotly (e.g., scatter plots on maps, choropleth maps, geographic heatmaps)?
3. Could you explain the data representation and source used within your program to create maps in Plotly?
4. Does your program showcase any interactive or customizable features available in Plotly for the created maps (e.g., zoom, hover information, custom styling)? If so, how are these features implemented?
5. How does your program handle potential errors or unexpected data inputs when creating different types of maps using Plotly?

Note : All Lab Programs from 4 to 10 can be written using files as input.such programs will be updated as earliest