**Matplotlib Basic Plotting**

1. **Simple Line Plot:** Create a NumPy array for x values (e.g., 1 to 5) and a corresponding array for y values (e.g., y=x2). Create a simple **line plot** of y versus x.
2. **Scatter Plot:** Using the same x and y arrays from Question 1, create a **scatter plot** of the data points.
3. **Title and Labels:** Add a **title** ("Simple Data Plot") and clear **labels** for the x-axis ("X-axis") and y-axis ("Y-axis") to your plot from Question 1.
4. **Customizing Color and Line Style:** Plot y versus x again, but set the line color to **red** and the line style to a **dashed line** (--).

**Customization and Annotations**

1. **Adding a Legend:** Plot two lines on the same axes: y1​=x and y2​=x2. Assign a **label** to each line and display a **legend**.
2. **Bar Chart:** Given two lists, categories = ['A', 'B', 'C'] and values = [10, 15, 7], create a **bar chart** showing the values for each category.
3. **Histogram:** Generate 100 random numbers from a standard normal distribution (np.random.randn(100)). Create a **histogram** of this data, showing the distribution.
4. **Setting Limits:** For the line plot from Question 1, explicitly set the **y-axis limits** to range from 0 to 30.
5. **Annotation:** Create a scatter plot of x and y. Add a text **annotation** at the point (3,9) that reads "Key Point".
6. **Customizing Markers:** Create a scatter plot where the points are drawn as **triangles** (^) and the size of the markers is set to 100.

**Subplots and Advanced Features**

1. **Simple Subplots (Rows and Cols):** Create a figure with a 1×2 grid of subplots. Plot the data y=x in the first subplot and y=x2 in the second subplot.
2. **Subplot Titles:** Add a specific title to each of the two subplots created in Question 11 (e.g., "Linear" and "Quadratic").
3. **Sharing Axes:** Create a 2×1 grid of subplots that **share the same x-axis** (sharex=True).
4. **Figure Size:** Create a single plot and explicitly set the size of the entire figure to be 10 inches wide and 5 inches tall.
5. **Adding Grid Lines:** Add a **grid** to a simple line plot to make the values easier to read.
6. **Pie Chart:** Given labels = ['East', 'West', 'North'] and sizes = [40, 30, 30], create a **pie chart**.

**Integration and Saving**

1. **Object-Oriented Plotting:** Create a figure and axes object explicitly using fig, ax = plt.subplots(). Use the axes object (ax) to plot y=x3 and set the title.
2. **Saving the Figure:** Create a simple line plot and use a function to **save the figure** to a file named my\_plot.png.
3. **Logarithmic Scale:** Plot y versus x. Change the **y-axis scale** to be **logarithmic** using ax.set\_yscale('log') (use the ax object from Question 17).
4. **Combining Plots (Conceptual):** Explain in one sentence the primary difference between using plt.plot() (from the procedural interface) and using ax.plot() (from the object-oriented interface).