

Part 1: Basics and Fundamentals (6 points)

1.1. Given a string `s="MasteringPythonIsRewarding"`, write a program to count the occurrence of each vowel in the string. (1 point)

Don't forget to count both uppercase and lowercase letters and we advice you to use the 'in' keyword to check if an element exists in the list as in the following example:.

```
my_list = [1, 2, 3, 4, 5]
if 3 in my_list:
    print("3 is in the list!")
```

1.2. Create a list called `squared_numbers` containing the squares of numbers from 1 to 15. (1 point)

1.3. Write a function `is_prime` that checks if a number is prime. Generate a list of primes between 1 and 50. (2 points)

We'd like to remind you that a prime number is a number divisible only by itself and one. We also advise you to use the Python integer division //

1.4. Create a function `string_alternate` that takes a string and returns a new string with every alternate character. (2 points)

For example, with the input 'PythonExam', the function should return 'PtoEa'

Part 2: Numpy and Intermediate Data Analysis (6 points)

2.1. Create a 4x4 numpy matrix `M` with numbers from 1 to 16. Print the matrix, then print the diagonal of `M`. (1 point)

2.2. Create two random 3x3 matrices `x` and `y`. Compute and print the matrix multiplication result of `x` and `y`. (1 point)

2.3. Generate an array `D` of 200 random numbers between -1 and 1 using numpy. Calculate and print the variance of `D`. (2 points)

2.4. Find and print the indices in `D` where the values are greater than 0.5. (1 point)

2.5. Convert the array `D` to have values rounded to 2 decimal places. (1 point)

Part 3: Pandas, Matplotlib, and yfinance (8 points)

3.1. Import the pandas library and create a DataFrame `df` from a dictionary with the following data:

```
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [25, 30, 35, 40],
    'Salary': [50000, 55000, 60000, 65000]
}
```

Print out the average age and average salary from the DataFrame. (2 points)

3.2. Using matplotlib, plot a bar graph that shows the salary of each individual from the DataFrame `df`. Ensure the graph has a title, x-axis label, y-axis label, and the names of the individuals on the x-axis. (2 points)

3.3. Import the `yfinance` library (if not already installed). Fetch the last 30 days' closing prices for the stock ticker "AAPL". Store this data in a DataFrame. (2 points)

3.4. Plot the closing prices of "AAPL" for the last 30 days using matplotlib. Label the x-axis as "Date" and the y-axis as "Closing Price". Add a title "AAPL 30-day Closing Price". (2 points)