```
import numpy as np
import pandas as pd

# NumPy Answers
# 1. Create a 1D array containing numbers from 0 to 9.
array_1d = np.arange(10)

# 2. Create a 3x3 matrix with values ranging from 0 to 8.
matrix_3x3 = np.arange(9).reshape(3, 3)

# 3. Create a 5x5 identity matrix.
identity_matrix = np.eye(5)

# 4. Create a 1D array with 10 equidistant values between 0 and 1.
equidistant_array = np.linspace(0, 1, 10)

# 5. Generate a random 3x3 matrix with values in the range [0, 1).
random_matrix = np.random.random((3, 3))

# 6. Multiply a 5x3 matrix by a 3x2 matrix.
matrix_a = np.random.random((5, 3))
matrix_b = np.random.random((3, 2))
matrix_multiply = np.dot(matrix_a, matrix_b)

# 7. Find the mean, median, and standard deviation of a given array.
array = np.random.random(100)
mean_value = np.mean(array)
median_value = np.median(array)
std_deviation = np.std(array)

# 8. Create a 3x3 matrix with random values and normalize it.
random_matrix = np.random.random((3, 3))
normalized_matrix = (random_matrix - random_matrix.mean()) /
random_matrix.std()
```

```
# 9. Reshape a 1D array into a 2D array with 2 rows.
reshaped_array = np.arange(10).reshape(2, -1)

# 10. Extract all odd numbers from an array.
odd_numbers = array[array % 2 != 0]

# 11. Reverse a given array (first element becomes last).
reversed_array = array[::-1]

# 12. Replace all odd numbers in an array with -1.
array[array % 2 != 0] = -1

# 13. Find the 5th and 95th percentile of a given array.
fifth_percentile = np.percentile(array, 5)
ninety_fifth_percentile = np.percentile(array, 95)

# 14. Compute the outer product of two given vectors.
vector_a = np.array([1, 2, 3])
vector_b = np.array([4, 5, 6])
outer_product = np.outer(vector_a, vector_b)

# 15. Compute the determinant of a given square matrix.
determinant = np.linalg.det(matrix_3x3)

# 16. Find the eigenvalues and eigenvectors of a given matrix.
eigenvalues, eigenvectors = np.linalg.eig(matrix_3x3)

# 17. Perform element-wise multiplication of two arrays.
element_wise_product = np.multiply(vector_a, vector_b)

# 18. Find the index of the maximum value in an array.
max_index = np.argmax(array)

# 19. Compute the inverse of a given matrix.
inverse_matrix = np.linalg.inv(matrix_3x3)
```

```
# 20. Calculate the dot product of two matrices.
dot_product = np.dot(matrix_a, matrix_b)

# Pandas Answers
# 21. Create a DataFrame from a dictionary of lists.
data = {'A': [1, 2, 3], 'B': [4, 5, 6], 'C': [7, 8, 9]}
df = pd.DataFrame(data)

# 22. Select the first 3 rows of a DataFrame.
first_three_rows = df.head(3)

# 23. Select a specific column from a DataFrame.
column_B = df['B']

# 24. Add a new column to an existing DataFrame.
df['D'] = [10, 11, 12]

# 25. Filter rows in a DataFrame based on a condition.
filtered_df = df[df['A'] > 1]

# 26. Group a DataFrame by a specific column and find the mean of each group.
grouped_mean = df.groupby('A').mean()

# 27. Merge two DataFrames based on a common column.
df1 = pd.DataFrame({'key': ['foo', 'bar', 'baz'], 'value': [1, 2, 3]})
df2 = pd.DataFrame({'key': ['foo', 'bar', 'baz'], 'value': [4, 5, 6]})
merged_df = pd.merge(df1, df2, on='key')

# 28. Rename the columns of a DataFrame.
df_renamed = df.rename(columns={'A': 'X', 'B': 'Y', 'C': 'Z'})

# 29. Sort a DataFrame by values in a specific column.
sorted_df = df.sort_values(by='B')
```

```
# 30. Pivot a DataFrame to reshape it based on column values.
pivot_df = pd.pivot_table(df, values='D', index='A', columns='B')

# Printing all results
print("NumPy Answers:")
print("1D Array:", array_1d)
print("3x3 Matrix:", matrix_3x3)
print("Identity Matrix:", identity_matrix)
print("Equidistant Array:", equidistant_array)
print("Random Matrix:", random_matrix)
print("Matrix Multiplication:", matrix_multiply)
print("Mean:", mean_value)
print("Median:", median_value)
print("Standard Deviation:", std_deviation)
print("Normalized Matrix:", normalized_matrix)
print("Reshaped Array:", reshaped_array)
print("Odd Numbers:", odd_numbers)
print("Reversed Array:", reversed_array)
print("Array with Odd Numbers Replaced:", array)
print("5th Percentile:", fifth_percentile)
print("95th Percentile:", ninety_fifth_percentile)
print("Outer Product:", outer_product)
print("Determinant:", determinant)
print("Eigenvalues:", eigenvalues)
print("Eigenvectors:", eigenvectors)
print("Element-wise Product:", element_wise_product)
print("Index of Maximum Value:", max_index)
print("Inverse Matrix:", inverse_matrix)
print("Dot Product:", dot_product)

print("\n\nPandas Answers:")
print("DataFrame from Dictionary:", df)
print("First 3 Rows:", first_three_rows)
print("Column B:", column_B)
print("DataFrame with New Column:", df)
```

```
print("Filtered DataFrame:", filtered_df)
print("Grouped Mean:", grouped_mean)
print("Merged DataFrame:", merged_df)
print("Renamed DataFrame:", df_renamed)
print("Sorted DataFrame:", sorted_df)
print("Pivoted DataFrame:", pivot_df)
```