Write 5 Numpy Programs:

```
import numpy as np

# 1. Generate Random Numbers Using `numpy.random.rand()`
random_data = np.random.rand(5)
print(f"Random Data: {random_data}")
```

Output:

Random Data: [0.65813452 0.24731543 0.73951217 0.12468211 0.55048393]

```
# 2. Calculate the Mean Using `numpy.mean()`
mean_val = np.mean(random_data)
print(f"Mean Value: {mean_val}")
```

Output:
Mean Value: 0.464025632

```
# 3. Reshape Array Using `numpy.reshape()`
reshaped_data = np.reshape(random_data, (5,1))
print(f"Reshaped Data:\n{reshaped_data}")
```

Output:
Reshaped Data:
[[0.65813452]
 [0.24731543]
 [0.73951217]
 [0.12468211]
 [0.55048393]]

```
# 4. Concatenate Arrays Using `numpy.concatenate()`
another_array = np.array([1, 2, 3, 4, 5])
concatenated_array = np.concatenate((random_data, another_array))
print(f"Concatenated Array: {concatenated_array}")
```

Concatenated Array: [0.65813452 0.24731543 0.73951217 0.12468211 0.55048393 1. 2. 3. 4. 5.]

 # 5. Compute the Cumulative Sum Using `numpy.cumsum()`

```
cumulative_sum = np.cumsum(random_data)
print(f"Cumulative Sum: {cumulative_sum}")
```

Cumulative Sum: [0.65813452 0.90544995 1.64496212 1.76964423 2.32012816]

Using Pandas read dataset and print it

```
import pandas as pd

# The URL for the "Iris" dataset in UCI repository
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

# Define the column names for the dataset
column_names = ["sepal_length", "sepal_width", "petal_length", "petal_width", "class"]

# Read the dataset into a pandas DataFrame
df = pd.read_csv(url, header=None, names=column_names)

# Show the first two lines
print(df.head(2))
```

Display two Matplot and Searborn chart using any dataset

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Load the iris dataset
iris = sns.load_dataset("iris")

# Plot using Matplotlib
plt.figure(figsize=(8, 6))
plt.scatter(iris["sepal_length"], iris["sepal_width"], c=iris["species"].astype("category").cat.codes)
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
```

```
plt.title("Sepal Length vs Sepal Width (Matplotlib)")
plt.show()

# Plot using Seaborn
plt.figure(figsize=(8, 6))
sns.scatterplot(data=iris, x="sepal_length", y="sepal_width", hue="species")
plt.title("Sepal Length vs Sepal Width (Seaborn)")
plt.show()
```

Write 5 Scipy functions:

```
import numpy
from scipy import special


#Create a numpy array
x = numpy.array([27,45])


#Get cube root
print(special.cbrt(x))


#Exponential function
print(special.exp10(x))


#Permutation and combination
print(special.comb(3,2)) #Equivalent to 3C2
print(special.perm(3,2)) #Equivalent to 3P2


#round function
print(special.round([1.9, -2.4, 1.2]))
```

Perform Data Analysis using house price prediction dataset

1. Loading the dataset:

```
import pandas as pd

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data"
column_names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
'PTRATIO', 'B', 'LSTAT', 'MEDV']

data = pd.read_csv(url, delim_whitespace=True, header=None, names=column_names)
```

2. Checking the first few rows:

```
print(data.head())
```

3. Summary statistics:

```
print(data.describe())
```

4. Checking for missing values:

```
print(data.isnull().sum())
```

5. Visualizing the data distribution:

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(rc={'figure.figsize':(11.7,8.27)})
sns.distplot(data['MEDV'], bins=30)
plt.show()
```

6. Correlation matrix:

```
correlation_matrix = data.corr().round(2)
sns.heatmap(data=correlation_matrix, annot=True)
```

7. Scatter plots for most important features:

```
plt.figure(figsize=(20, 5))

features = ['LSTAT', 'RM']
target = data['MEDV']

for i, col in enumerate(features):
    plt.subplot(1, len(features), i+1)
    x = data[col]
    y = target
    plt.scatter(x, y, marker='o')
    plt.title(col)
    plt.xlabel(col)
    plt.ylabel('MEDV')
```

Scrape data from AMAZON website

```
import requests
from bs4 import BeautifulSoup

# Set the starting URL
url = "https://www.amazon.com/s?k=electronics&page=1"

# Create a list to store the scraped data
scraped_data = []
```

```
# Iterate over the pages
for i in range(1, 1001):
    # Make a request to the Amazon website
    response = requests.get(url)

    # Parse the HTML response using BeautifulSoup
    soup = BeautifulSoup(response.content, "html.parser")

    # Extract the product data from the HTML
    for product in soup.find_all("div", class_="sg-col-4-of-12 s-result-item s-asin sg-col-4-of-16
s-widget-spacing-small sg-col-4-of-20"):
        product_name = product.find("span", class_="a-size-base-plus a-color-base").text
        product_price = product.find("span", class_="a-price-whole").text
        product_rating = product.find("span", class_="a-icon-alt").text

        # Add the product data to the scraped_data list
        scraped_data.append({
            "product_name": product_name,
            "product_price": product_price,
            "product_rating": product_rating,
        })

    # Update the URL to the next page
    url = url.replace("page={}".format(i), "page={}".format(i + 1))

# Save the scraped data to a CSV file
import pandas as pd

df = pd.DataFrame(scraped_data)
df.to_csv("amazon_products.csv", index=False)
```