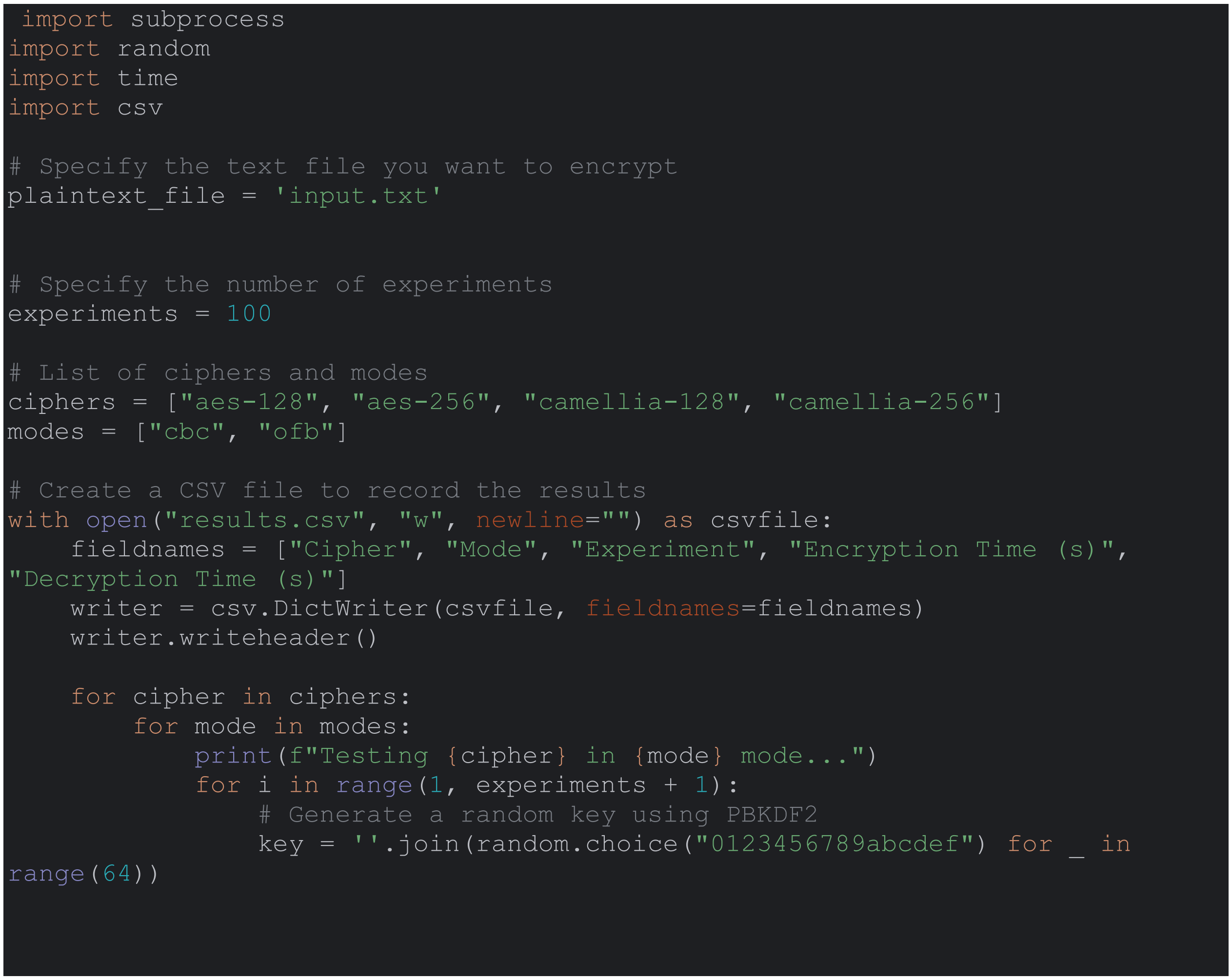
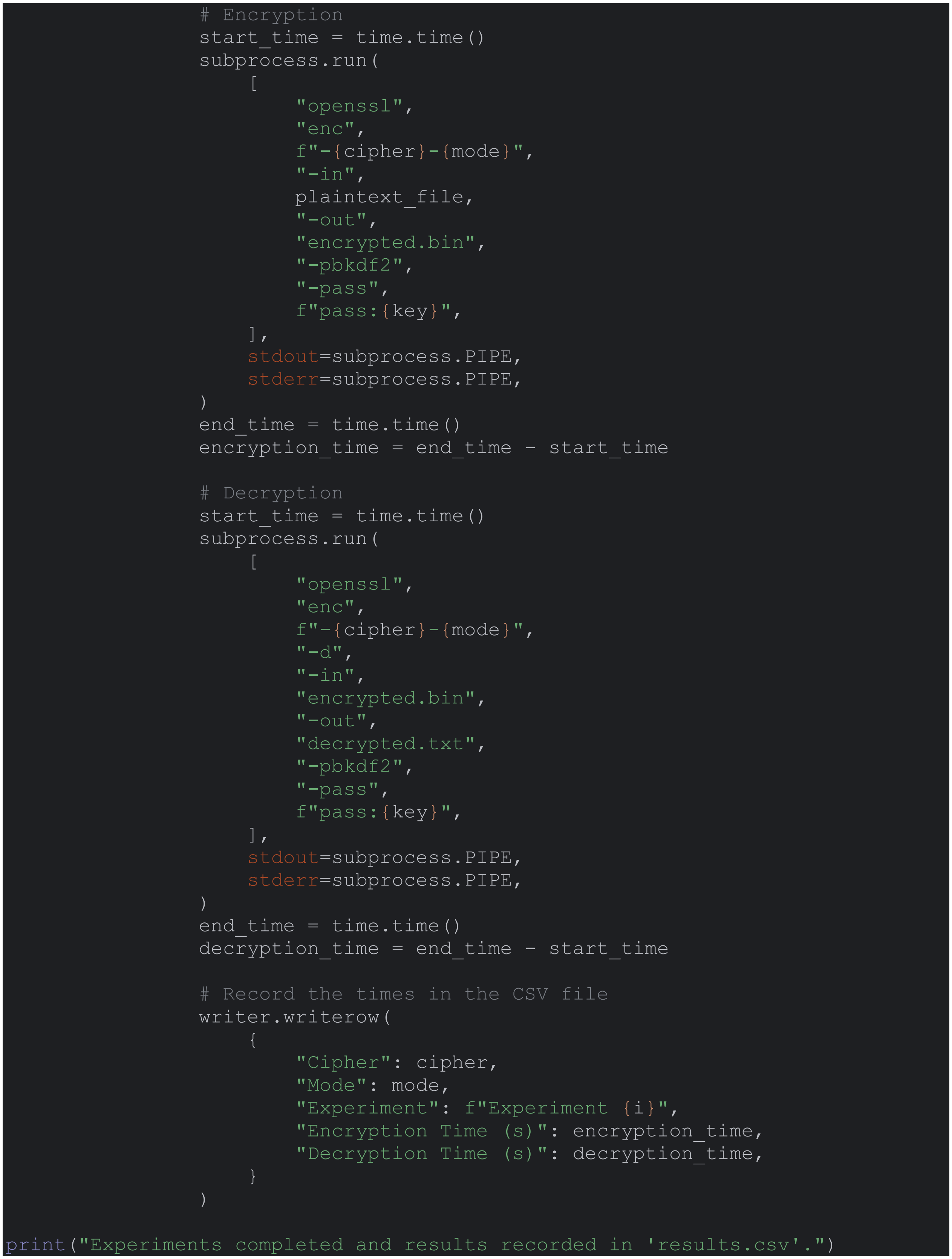
1) 4 ciphers that I used :
    1) E1 : AES 128
    2) E2 : AES 256
    3) E3 : camellia 128
    4) E4 : camellia 256

2) two modes of operations :
    1) B1 : CBC
    2) B2 : OFB

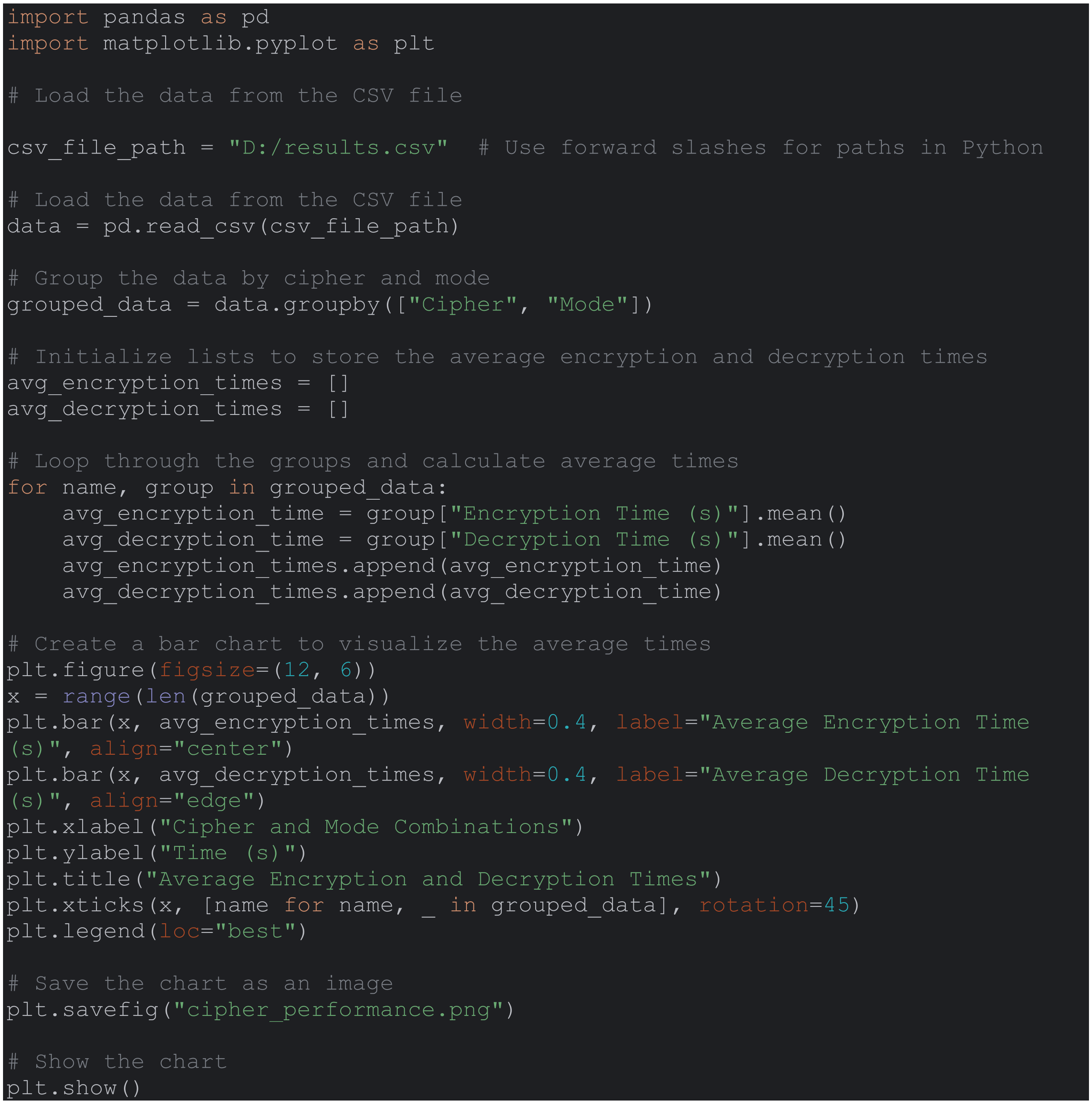
My code in Python :



```
 import subprocess
import random
import time
import csv

# Specify the text file you want to encrypt
plaintext_file = 'input.txt'


# Specify the number of experiments
experiments = 100

# List of ciphers and modes
ciphers = ["aes-128", "aes-256", "camellia-128", "camellia-256"]
modes = ["cbc", "ofb"]

# Create a CSV file to record the results
with open("results.csv", "w", newline="") as csvfile:
    fieldnames = ["Cipher", "Mode", "Experiment", "Encryption Time (s)",
"Decryption Time (s)"]
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()

    for cipher in ciphers:
        for mode in modes:
            print(f"Testing {cipher} in {mode} mode...")
            for i in range(1, experiments + 1):
                # Generate a random key using PBKDF2
                key = ''.join(random.choice("0123456789abcdef") for _ in
range(64))
```

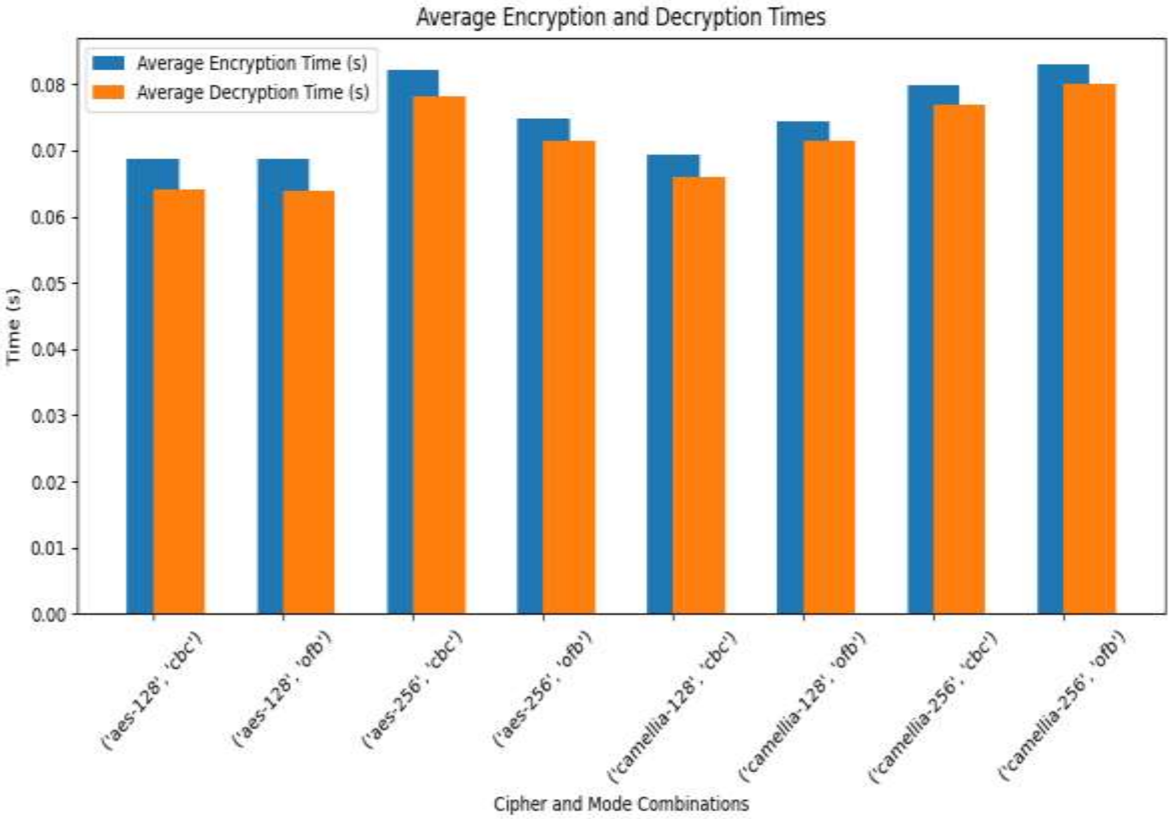


```
                # Encryption
                start_time = time.time()
                subprocess.run(
                    [
                        "openssl",
                        "enc",
                        f"-{cipher}-{mode}",
                        "-in",
                        plaintext_file,
                        "-out",
                        "encrypted.bin",
                        "-pbkdf2",
                        "-pass",
                        f"pass:{key}",
                    ],
                    stdout=subprocess.PIPE,
                    stderr=subprocess.PIPE,
                )
                end_time = time.time()
                encryption_time = end_time - start_time

                # Decryption
                start_time = time.time()
                subprocess.run(
                    [
                        "openssl",
                        "enc",
                        f"-{cipher}-{mode}",
                        "-d",
                        "-in",
                        "encrypted.bin",
                        "-out",
                        "decrypted.txt",
                        "-pbkdf2",
                        "-pass",
                        f"pass:{key}",
                    ],
                    stdout=subprocess.PIPE,
                    stderr=subprocess.PIPE,
                )
                end_time = time.time()
                decryption_time = end_time - start_time

                # Record the times in the CSV file
                writer.writerow(
                    {
                        "Cipher": cipher,
                        "Mode": mode,
                        "Experiment": f"Experiment {i}",
                        "Encryption Time (s)": encryption_time,
                        "Decryption Time (s)": decryption_time,
                    }
                )

print("Experiments completed and results recorded in 'results.csv'.")
```



```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the data from the CSV file

csv_file_path = "D:/results.csv"  # Use forward slashes for paths in Python

# Load the data from the CSV file
data = pd.read_csv(csv_file_path)

# Group the data by cipher and mode
grouped_data = data.groupby(["Cipher", "Mode"])

# Initialize lists to store the average encryption and decryption times
avg_encryption_times = []
avg_decryption_times = []

# Loop through the groups and calculate average times
for name, group in grouped_data:
    avg_encryption_time = group["Encryption Time (s)"].mean()
    avg_decryption_time = group["Decryption Time (s)"].mean()
    avg_encryption_times.append(avg_encryption_time)
    avg_decryption_times.append(avg_decryption_time)

# Create a bar chart to visualize the average times
plt.figure(figsize=(12, 6))
x = range(len(grouped_data))
plt.bar(x, avg_encryption_times, width=0.4, label="Average Encryption Time
(s)", align="center")
plt.bar(x, avg_decryption_times, width=0.4, label="Average Decryption Time
(s)", align="edge")
plt.xlabel("Cipher and Mode Combinations")
plt.ylabel("Time (s)")
plt.title("Average Encryption and Decryption Times")
plt.xticks(x, [name for name, _ in grouped_data], rotation=45)
plt.legend(loc="best")

# Save the chart as an image
plt.savefig("cipher_performance.png")

# Show the chart
plt.show()
```

Perform for one hundred experiments per pair Ei, Bj (the results by drawings) :


Average Encryption and Decryption Times
Average Encryption Time (s)
Average Decryption Time (s)
Time (s)
('aes-128', 'cbc')
('aes-128', 'ofb')
('aes-256', 'cbc')
('aes-256', 'ofb')
('camellia-128', 'cbc')
('camellia-128', 'ofb')
('camellia-256', 'cbc')
('camellia-256', 'ofb')
Cipher and Mode Combinations