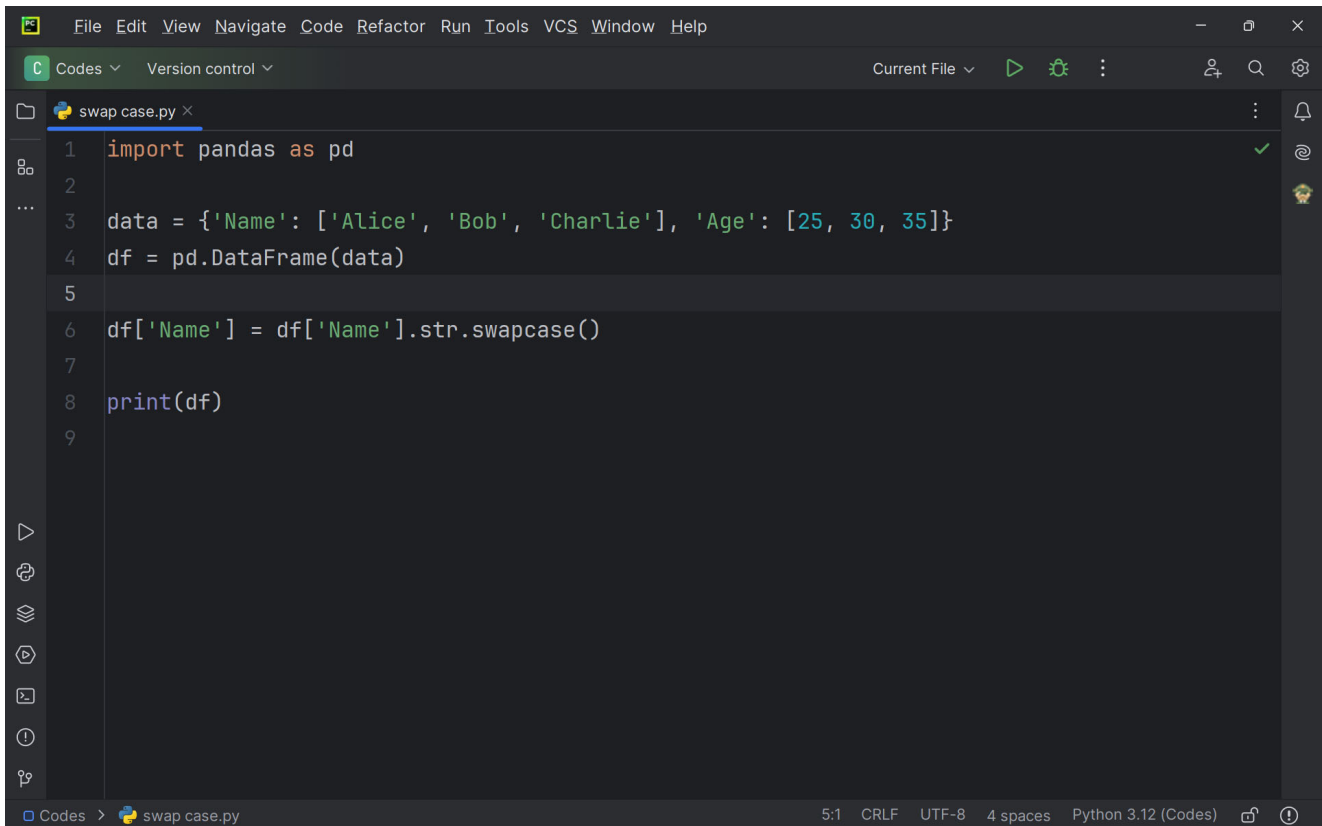


Experiment 21

Aim:

To develop a Pandas program to swap the cases of a character of a specified column.

Code:

A screenshot of a code editor interface. The top menu bar includes 'File', 'Edit', 'View', 'Navigate', 'Code', 'Refactor', 'Run', 'Tools', 'VCS', 'Window', and 'Help'. Below the menu is a toolbar with icons for running, debugging, and other functions. The main editor area shows a file named 'swap case.py' with the following Python code:

```
1 import pandas as pd
2
3 data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}
4 df = pd.DataFrame(data)
5
6 df['Name'] = df['Name'].str.swapcase()
7
8 print(df)
9
```

The status bar at the bottom indicates '5:1 CRLF UTF-8 4 spaces Python 3.12 (Codes)'.

Input:

```
data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}
df = pd.DataFrame(data)

df['Name'] = df['Name'].str.swapcase()
```

Output:

Run swap case x

↺

↑

↓

⇄

⇅

🗑

```
"C:\Users\maano_0waenfu\OneDrive\College\Query Processing\Codes\.venv\Scripts\python.exe" "C:\Us

      Name  Age
0    aLICE   25
1    bOB    30
2  cHARLIE   35

Process finished with exit code 0
|
```

Experiment 22

Aim:

To develop a Pandas program to plot a straight line.

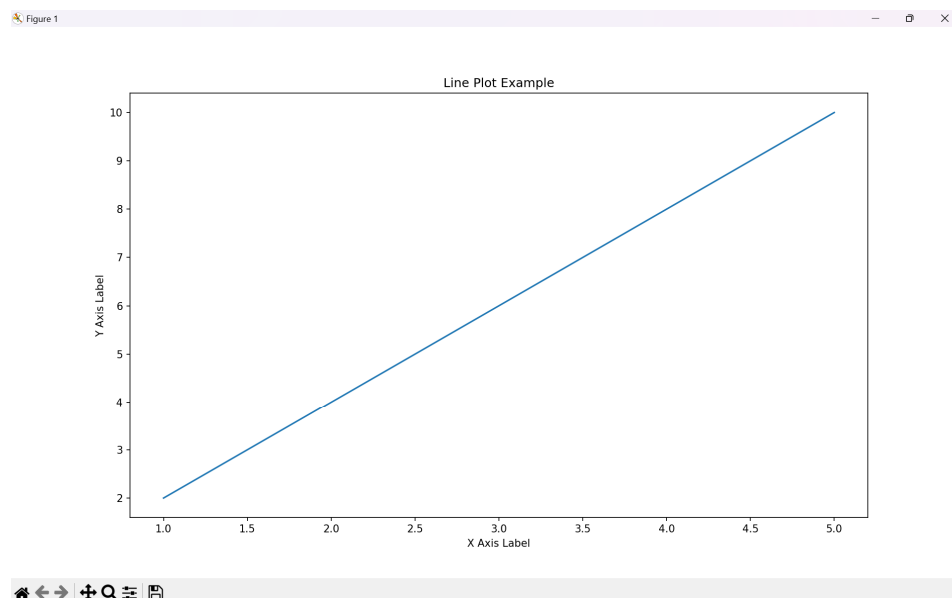
Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Codes Version control Current File
line plot.py
1 import matplotlib.pyplot as plt
2
3 x = [1, 2, 3, 4, 5]
4 y = [2, 4, 6, 8, 10]
5
6 plt.plot(*args: x, y)
7
8 # Add labels and title
9 plt.xlabel('X Axis Label')
10 plt.ylabel('Y Axis Label')
11 plt.title('Line Plot Example')
12
13 plt.show()
14
```

Input:

```
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
```

Output:

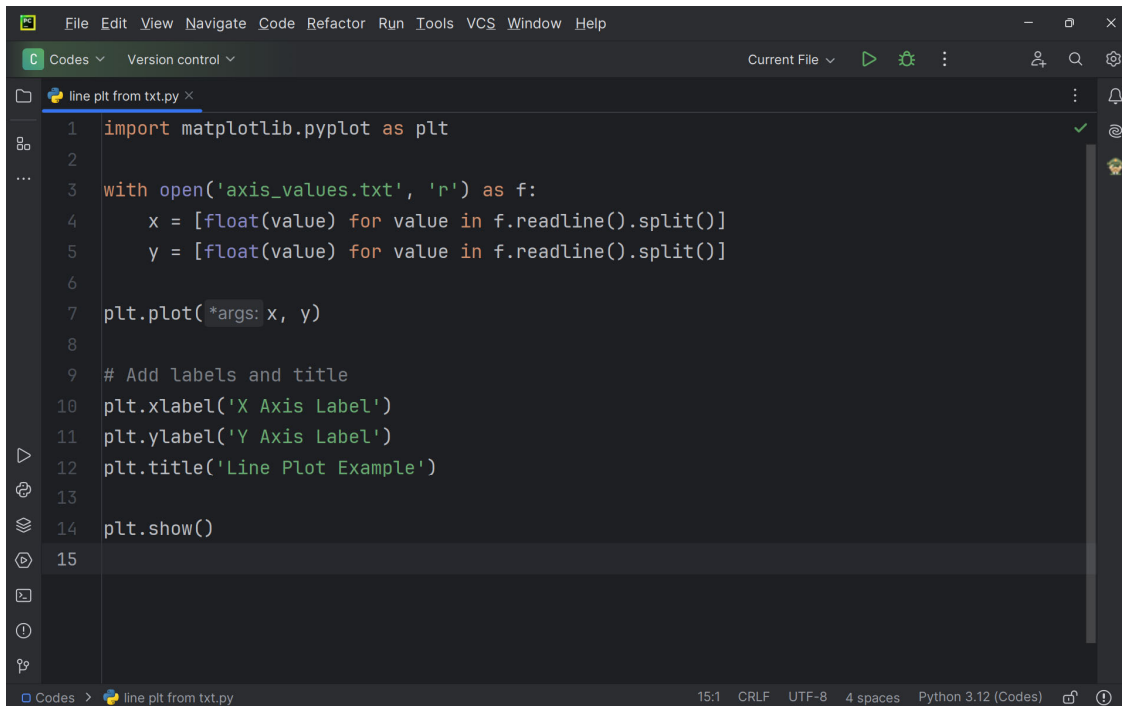


Experiment 23

Aim:

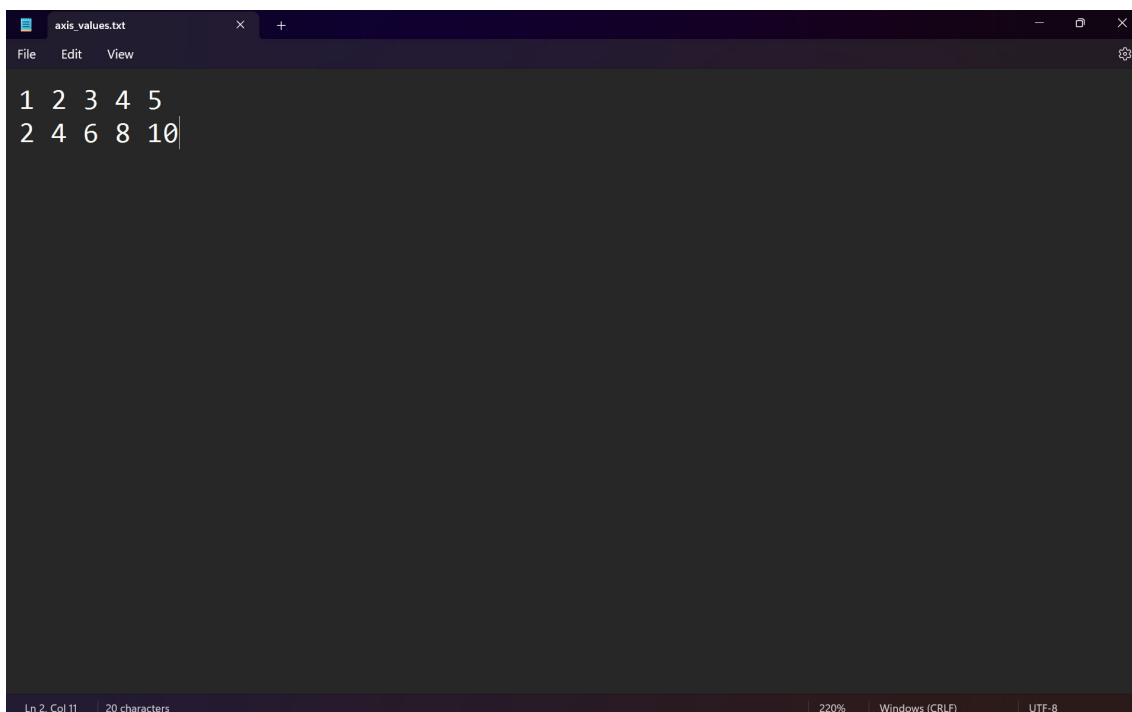
To develop a Pandas program to plot a straight-line taking axes values in .txt file.

Code:



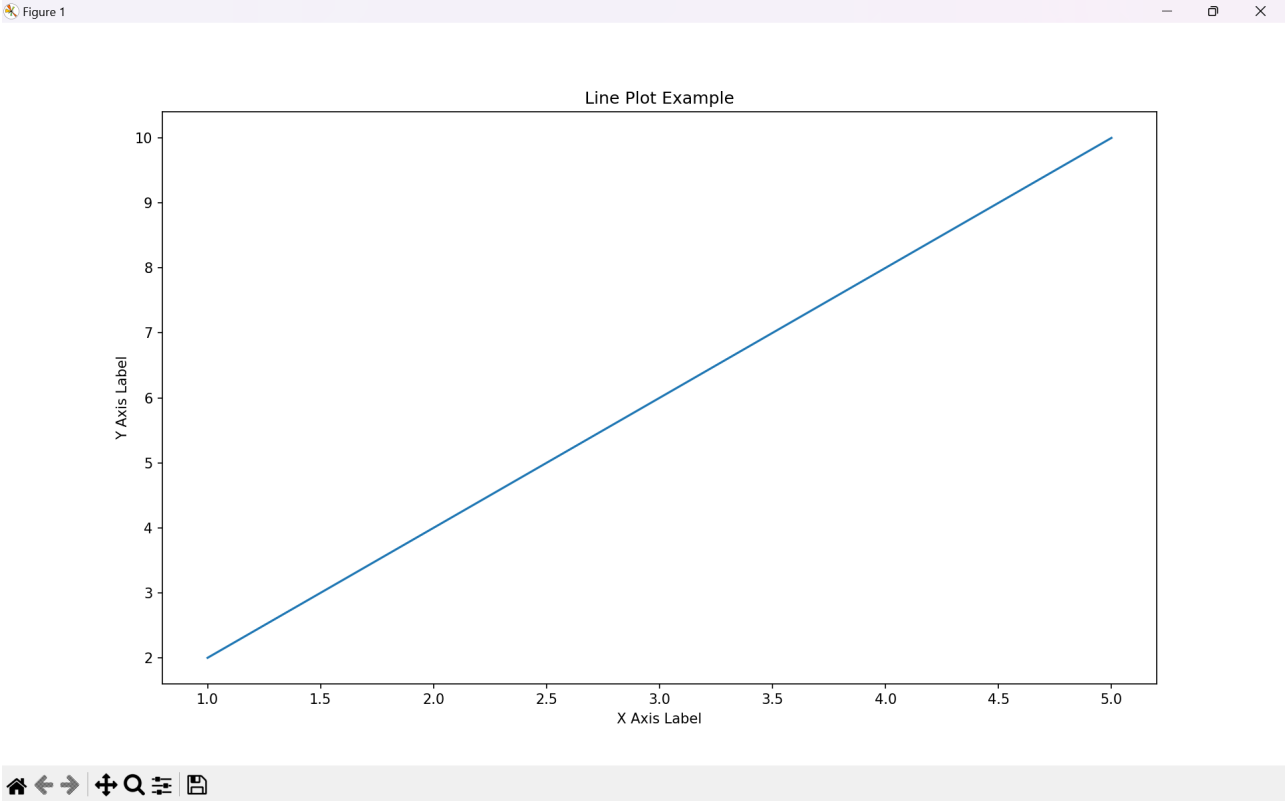
```
1 import matplotlib.pyplot as plt
2
3 with open('axis_values.txt', 'r') as f:
4     x = [float(value) for value in f.readline().split()]
5     y = [float(value) for value in f.readline().split()]
6
7 plt.plot(*args: x, y)
8
9 # Add labels and title
10 plt.xlabel('X Axis Label')
11 plt.ylabel('Y Axis Label')
12 plt.title('Line Plot Example')
13
14 plt.show()
15
```

Input:



```
axis_values.txt
1 2 3 4 5
2 4 6 8 10
```

Output:



Experiment 24

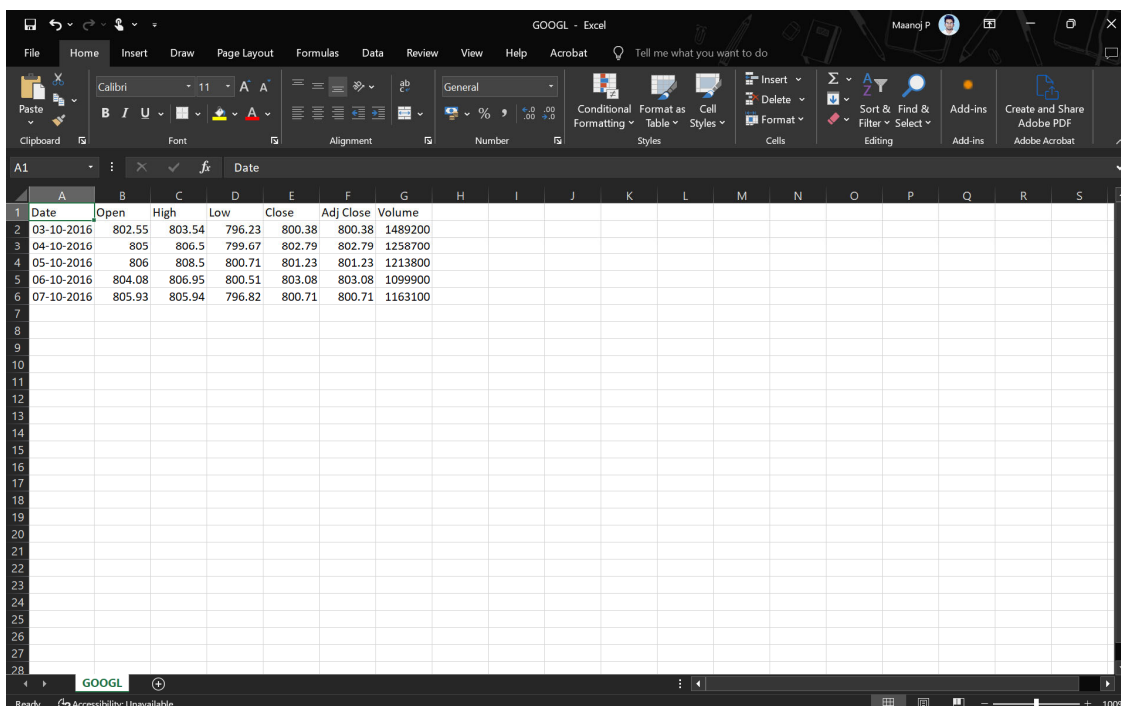
Aim:

To develop a Pandas program to plot a line chart for GOOGL financial dataset.

Code:

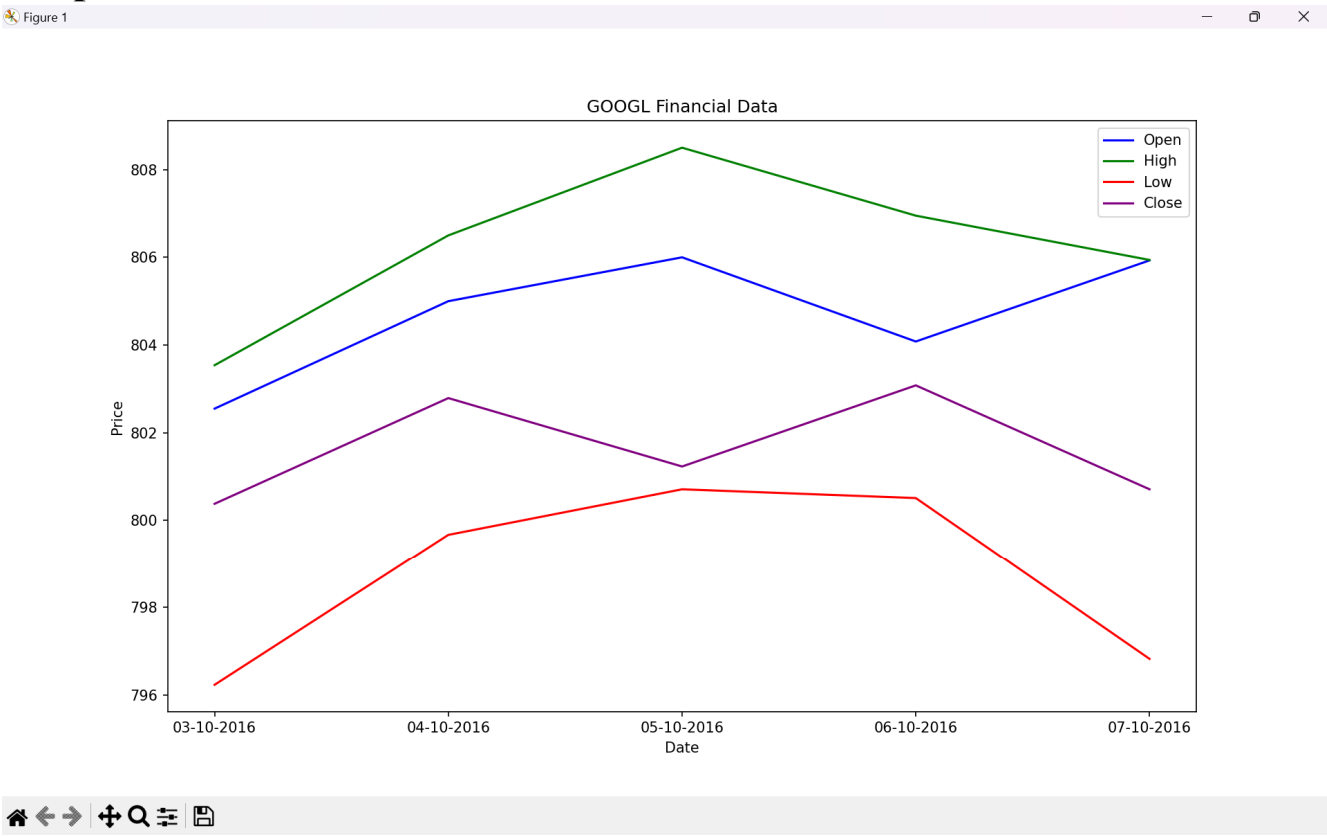
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Codes Version control Current File
line chart fin data.py
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 df = pd.read_csv('GOOGL.csv')
5
6
7 plt.figure(figsize=(14, 7))
8 plt.plot(*args: df['Date'], df['Open'], label='Open', color='blue')
9 plt.plot(*args: df['Date'], df['High'], label='High', color='green')
10 plt.plot(*args: df['Date'], df['Low'], label='Low', color='red')
11 plt.plot(*args: df['Date'], df['Close'], label='Close', color='purple')
12
13 plt.title('GOOGL Financial Data')
14 plt.xlabel('Date')
15 plt.ylabel('Price')
16 plt.legend()
17 plt.show()
18
19
Codes > line chart fin data.py 15:20 CRLF UTF-8 4 spaces Python 3.12 (Codes)
```

Input:



Date	Open	High	Low	Close	Adj Close	Volume
03-10-2016	802.55	803.54	796.23	800.38	800.38	1489200
04-10-2016	805	806.5	799.67	802.79	802.79	1258700
05-10-2016	806	808.5	800.71	801.23	801.23	1213800
06-10-2016	804.08	806.95	800.51	803.08	803.08	1099900
07-10-2016	805.93	805.94	796.82	800.71	800.71	1163100

Output:



Experiment 25

Aim:

To develop a Pandas program to plot a line chart with 2 or more values

Code:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Codes Version control Current File
2 or more line plot.py
1 import matplotlib.pyplot as plt
2
3 x = [1, 2, 3, 4, 5]
4 y1 = [1, 4, 9, 16, 25]
5 y2 = [2, 5, 8, 11, 14]
6
7 plt.plot(*args: x, y1, label='Line 1', color='red', linewidth=2)
8 plt.plot(*args: x, y2, label='Line 2', color='blue', linewidth=4)
9
10 plt.legend()
11
12 plt.title('Multiple Lines Plot')
13 plt.xlabel('X')
14 plt.ylabel('Y')
15
16 plt.show()
Codes 2 or more line plot.py 3:20 CRLF UTF-8 4 spaces Python 3.12 (Codes)
```

Input:

```
x = [1, 2, 3, 4, 5]
y1 = [1, 4, 9, 16, 25]
```

Output:

