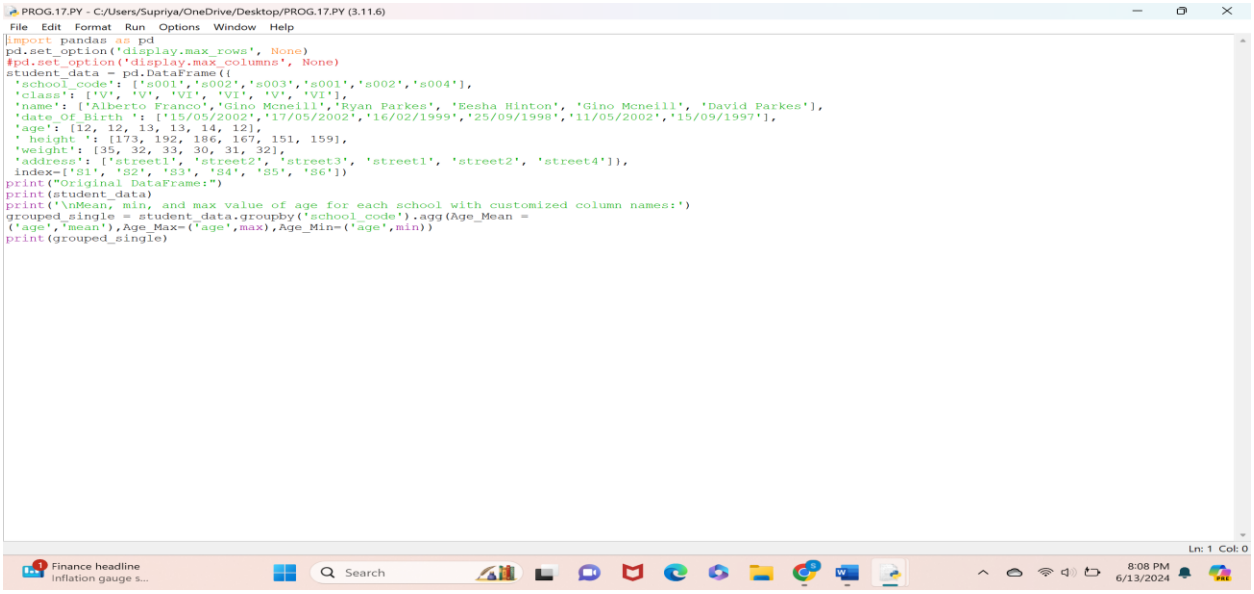
17.Write a Pandas program to split the following dataframe by school code and get mean, min, and max value of age for each school.

	school	class	name	date_Of_Birth	age	height	weight	address
S1	s001	V	Alberto Franco	15/05/2002	12	173	35	street1
S2	s002	V	Gino Mcneill	17/05/2002	12	192	32	street2
S3	s003	VI	Ryan Parkes	16/02/1999	13	186	33	street3
S4	s001	VI	Eesha Hinton	25/09/1998	13	167	30	street1
S5	s002	V	Gino Mcneill	11/05/2002	14	151	31	street2
S6	s004	VI	David Parkes	15/09/1997	12	159	32	street4

INPUT:



OUTPUT:

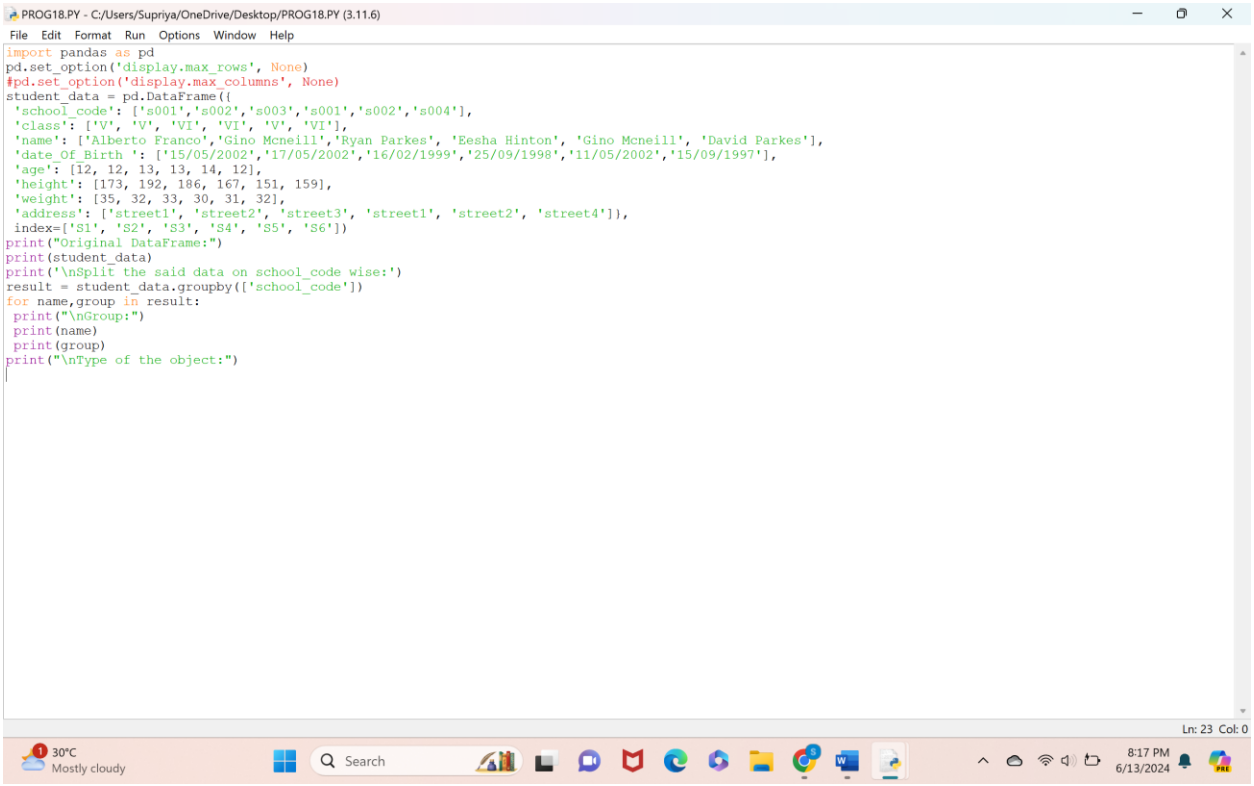
```
Original DataFrame:
   school_code class            name  ...  height   weight  address
S1        s001     V  Alberto Franco  ...     173      35  street1
S2        s002     V    Gino Mcneill  ...     192      32  street2
S3        s003    VI     Ryan Parkes  ...     186      33  street3
S4        s001    VI    Eesha Hinton  ...     167      30  street1
S5        s002     V    Gino Mcneill  ...     151      31  street2
S6        s004    VI    David Parkes  ...     159      32  street4

[6 rows x 8 columns]

Mean, min, and max value of age for each school with customized column names:
```

```
              Age_Mean  Age_Max  Age_Min
school_code
s001              12.5       13       12
s002              13.0       14       12
s003              13.0       13       13
s004              12.0       12       12
```

18.Write a Pandas program to split the following given dataframe into groups based on school code and class.

	school	class	name	date_Of_Birth	age	height	weight	address
S1	s001	V	Alberto Franco	15/05/2002	12	173	35	street1
S2	s002	V	Gino Mcneill	17/05/2002	12	192	32	street2
S3	s003	VI	Ryan Parkes	16/02/1999	13	186	33	street3
S4	s001	VI	Eesha Hinton	25/09/1998	13	167	30	street1
S5	s002	V	Gino Mcneill	11/05/2002	14	151	31	street2
S6	s004	VI	David Parkes	15/09/1997	12	159	32	street4

INPUT:



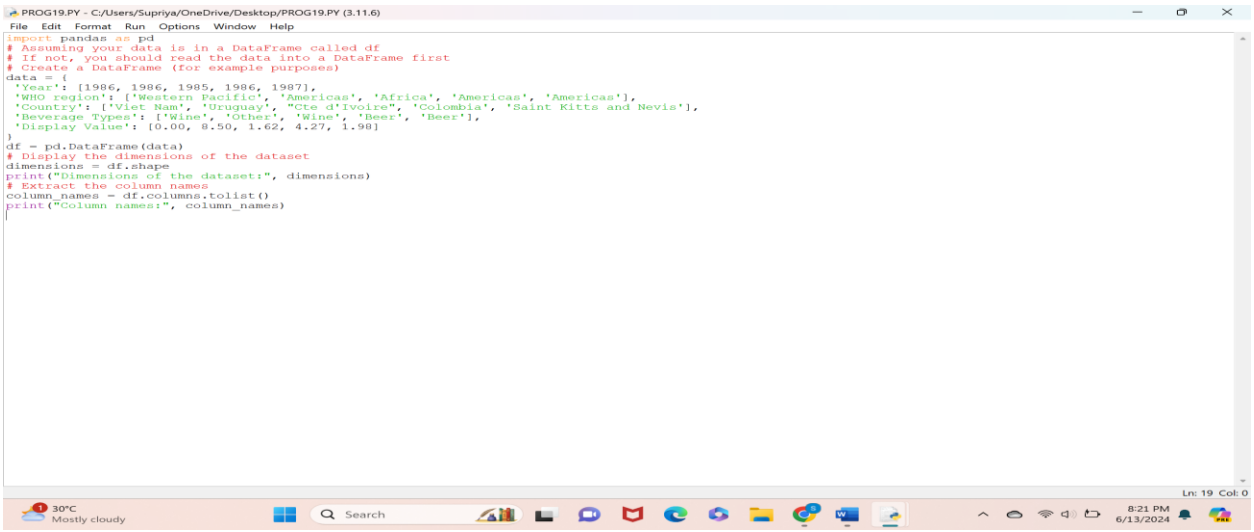
OUTPUT:

```
= RESTART: C:/Users/Supriya/OneDrive/Desktop/PROG18.PY
Original DataFrame:
   school_code class            name  ... height  weight  address
S1        s001     V  Alberto Franco  ...    173      35  street1
S2        s002     V    Gino Mcneill  ...    192      32  street2
S3        s003    VI     Ryan Parkes  ...    186      33  street3
S4        s001    VI    Eesha Hinton  ...    167      30  street1
S5        s002     V    Gino Mcneill  ...    151      31  street2
S6        s004    VI    David Parkes  ...    159      32  street4

[6 rows x 8 columns]

Split the said data on school_code wise:

Group:
('s001',)
   school_code class            name  ... height  weight  address
S1        s001     V  Alberto Franco  ...    173      35  street1
S4        s001    VI    Eesha Hinton  ...    167      30  street1

[2 rows x 8 columns]

Group:
('s002',)
   school_code class          name date_Of_Birth   age  height  weight  address
S2        s002     V  Gino Mcneill     17/05/2002   12     192      32  street2
S5        s002     V  Gino Mcneill     11/05/2002   14     151      31  street2

Group:
('s003',)
   school_code class         name date_Of_Birth   age  height  weight  address
S3        s003    VI  Ryan Parkes     16/02/1999   13     186      33  street3

Group:
('s004',)
   school_code class          name date_Of_Birth   age  height  weight  address
S6        s004    VI  David Parkes     15/09/1997   12     159      32  street4
```

19.Write a Pandas program to display the dimensions or shape of the World alcohol consumption dataset. Also extract the column names from the dataset.

	Year	WHO region	Country	Beverage Types	Display Value
0	1986	Western Pacific	Viet Nam	Wine	0.00
1	1986	Americas	Uruguay	Other	0.50
2	1985	Africa	Cte d'Ivoire	Wine	1.62
3	1986	Americas	Colombia	Beer	4.27
4	1987	Americas	Saint Kitts and Nevis	Beer	1.98

INPUT:

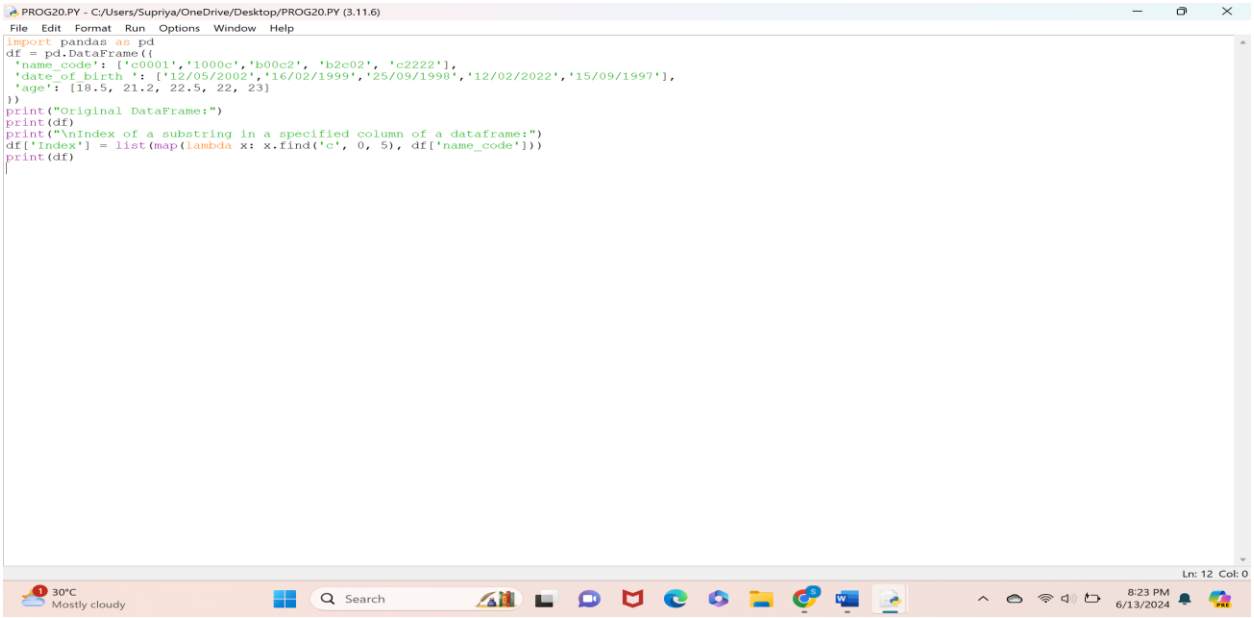


OUTPUT:

```
Dimensions of the dataset: (5, 5)
Column names: ['Year', 'WHO region', 'Country', 'Beverage Types', 'Display Value']
```

20.Write a Pandas program to find the index of a given substring of a DataFrame column.

INPUT:

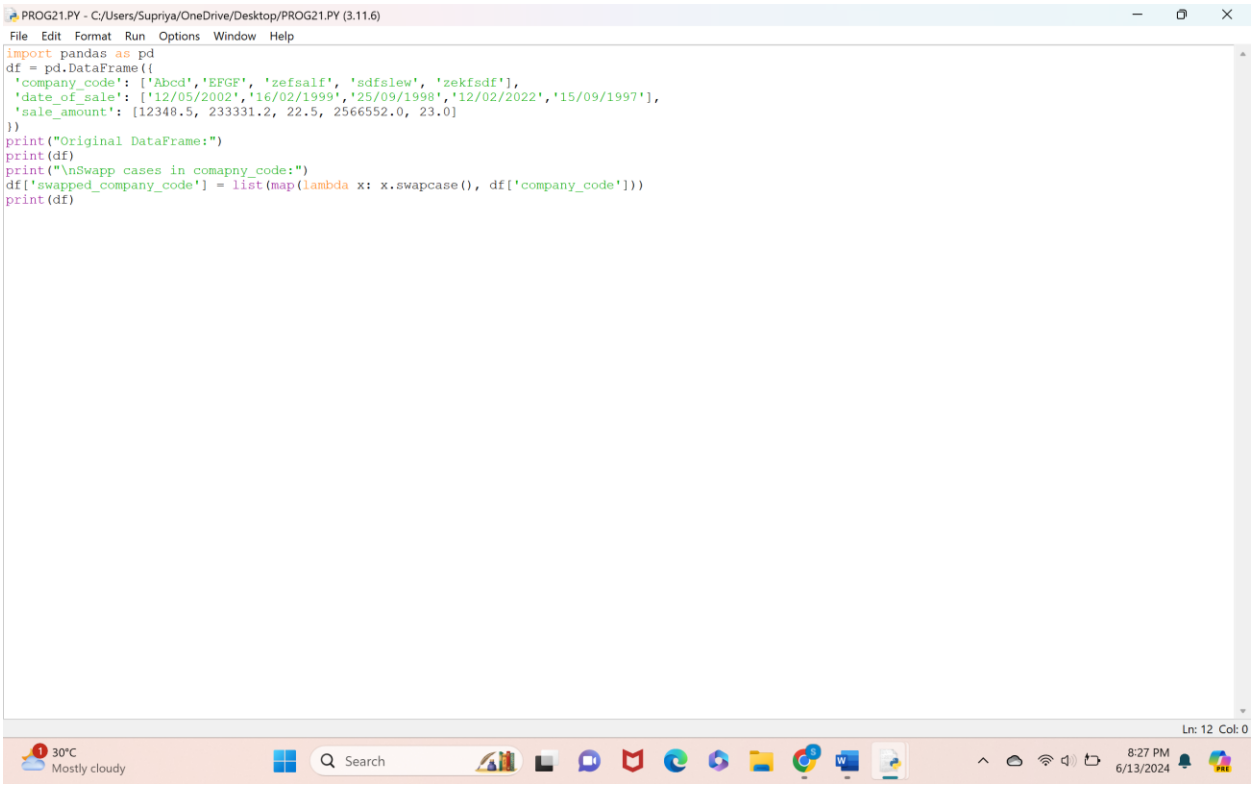


OUTPUT:

```
Original DataFrame:
  name_code date_of_birth    age
0     c0001      12/05/2002  18.5
1     1000c      16/02/1999  21.2
2     b00c2      25/09/1998  22.5
3     b2c02      12/02/2022  22.0
4     c2222      15/09/1997  23.0

Index of a substring in a specified column of a dataframe:
  name_code date_of_birth    age  Index
0     c0001      12/05/2002  18.5      0
1     1000c      16/02/1999  21.2      4
2     b00c2      25/09/1998  22.5      3
3     b2c02      12/02/2022  22.0      2
4     c2222      15/09/1997  23.0      0
```

21.Write a Pandas program to swap the cases of a specified character column in a given DataFrame.

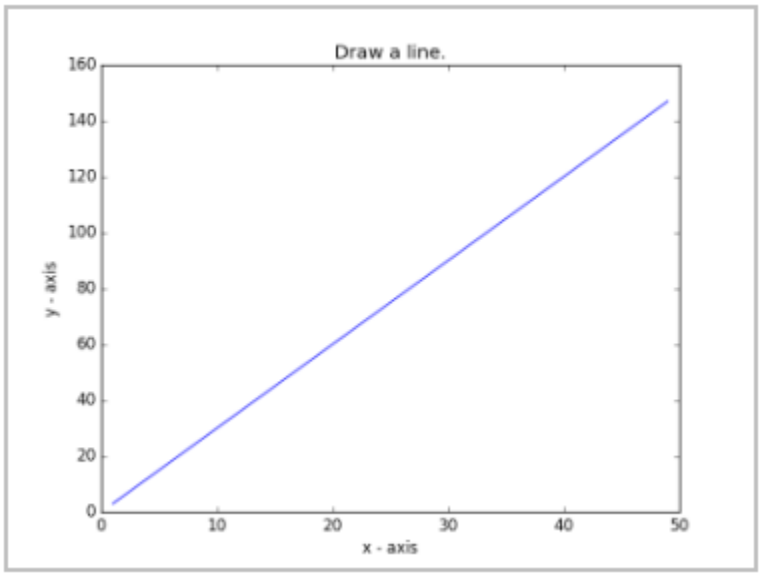
INPUT:



OUTPUT:

```
Original DataFrame:
  company_code date_of_sale  sale_amount
0         Abcd   12/05/2002      12348.5
1         EFGF   16/02/1999     233331.2
2      zefsalf   25/09/1998         22.5
3      sdfslew   12/02/2022    2566552.0
4      zekfsdf   15/09/1997         23.0

Swapp cases in comapny_code:
  company_code date_of_sale  sale_amount swapped_company_code
0         Abcd   12/05/2002      12348.5                  aBCD
1         EFGF   16/02/1999     233331.2                  efgf
2      zefsalf   25/09/1998         22.5               ZEFSALF
3      sdfslew   12/02/2022    2566552.0               SDFSLEW
4      zekfsdf   15/09/1997         23.0               ZEKFSDF
```

22.Write a Python program to draw a line with suitable label in the x axis, y axis and a title.



INPUT:

PROG22.PY - C:/Users/Supriya/OneDrive/Desktop/PROG22.PY (3.11.6)

File Edit Format Run Options Window Help

```
import matplotlib.pyplot as plt
X = range(1, 50)
Y = [value * 3 for value in X]
print("Values of X:")
print(*range(1,50))
print("Values of Y (thrice of X):")
print(Y)
# Plot lines and/or markers to the Axes.
plt.plot(X, Y)
# Set the x axis label of the current axis.
plt.xlabel('x - axis')
# Set the y axis label of the current axis.
plt.ylabel('y - axis')
# Set a title
plt.title('Draw a line.')
# Display the figure.
plt.show()
```

Ln: 18 Col: 0

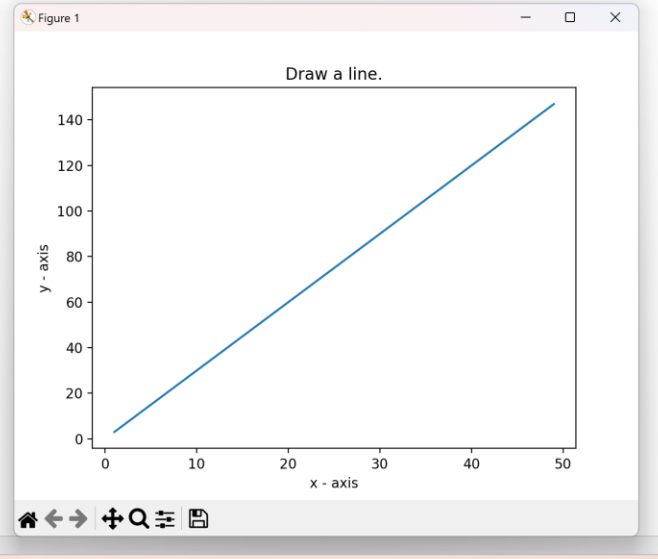
Search 8:30 PM 6/13/2024

OUTPUT:

IDLE Shell 3.11.6

File Edit Shell Debug Options Window Help

```
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Supriya/OneDrive/Desktop/PROG22.PY
Values of X:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
Values of Y (thrice of X):
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114
, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147]
```



Ln: 5 Col: 0

30°C Mostly cloudy Search 8:30 PM 6/13/2024

23.Write a Python program to draw a line using given axis values taken from a text file, with suitable label in the x axis, y axis and a title.

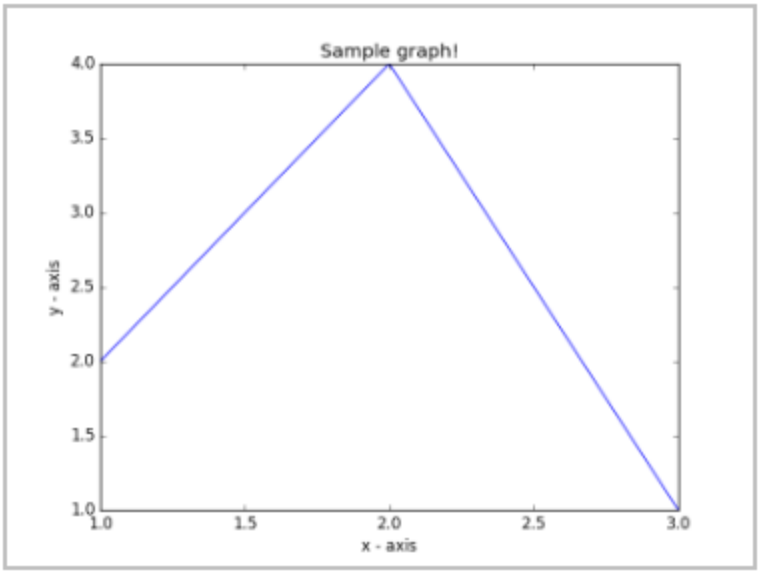
Test Data:

test.txt

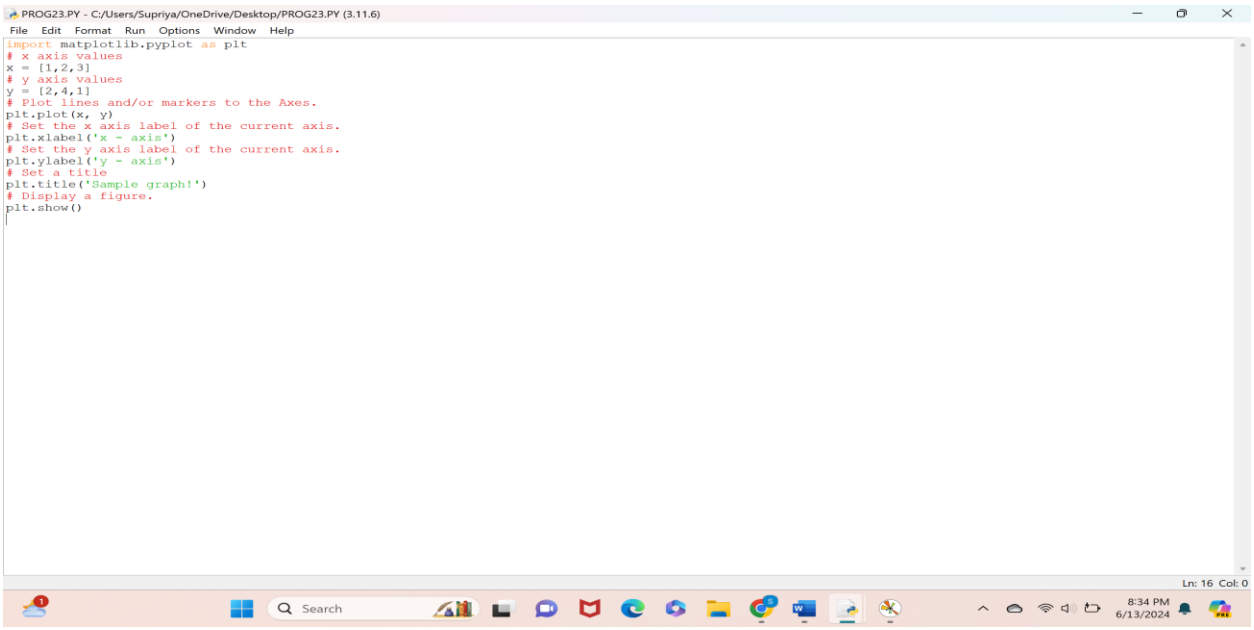
1 2

2 4

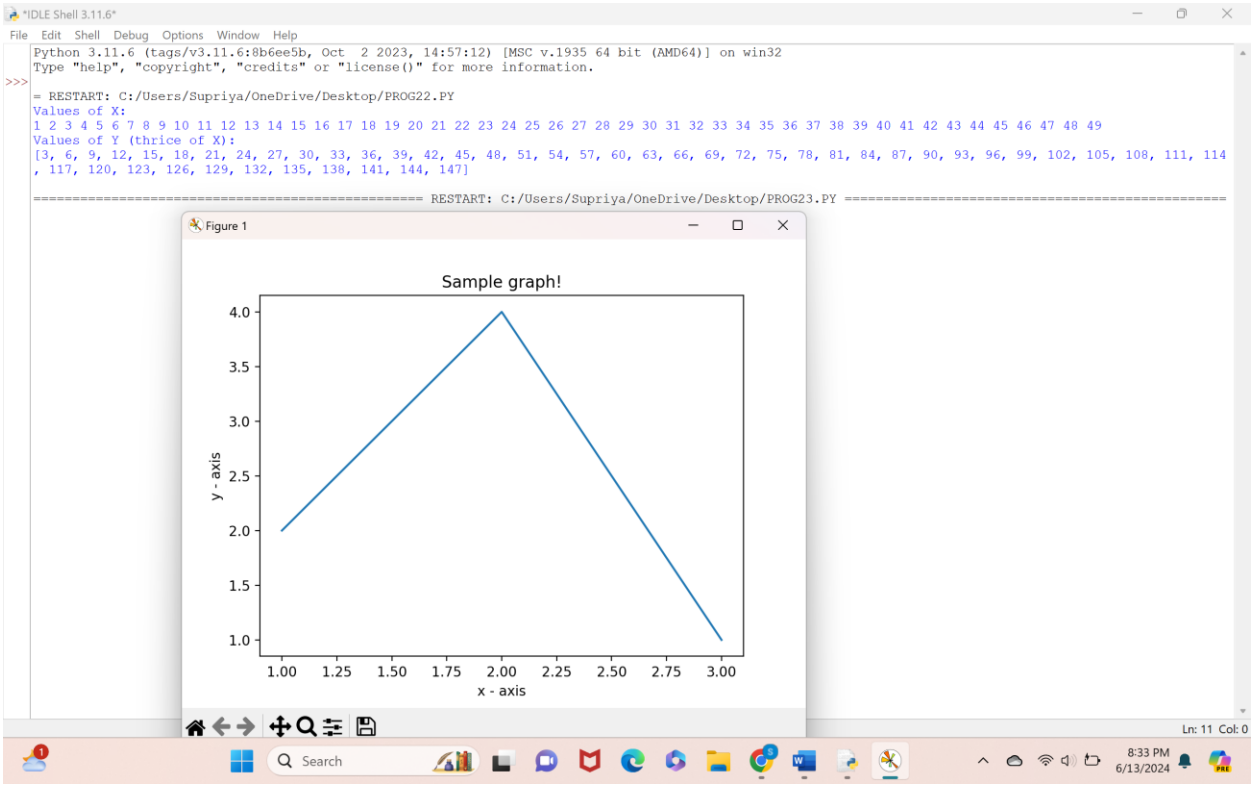
3 1



INPUT:



OUTPUT:



24.Write a Python program to draw line charts of the financial data of Alphabet Inc. between October 3, 2016 to October 7, 2016.

Sample Financial data (fdata.csv):

Date,Open,High,Low,Close

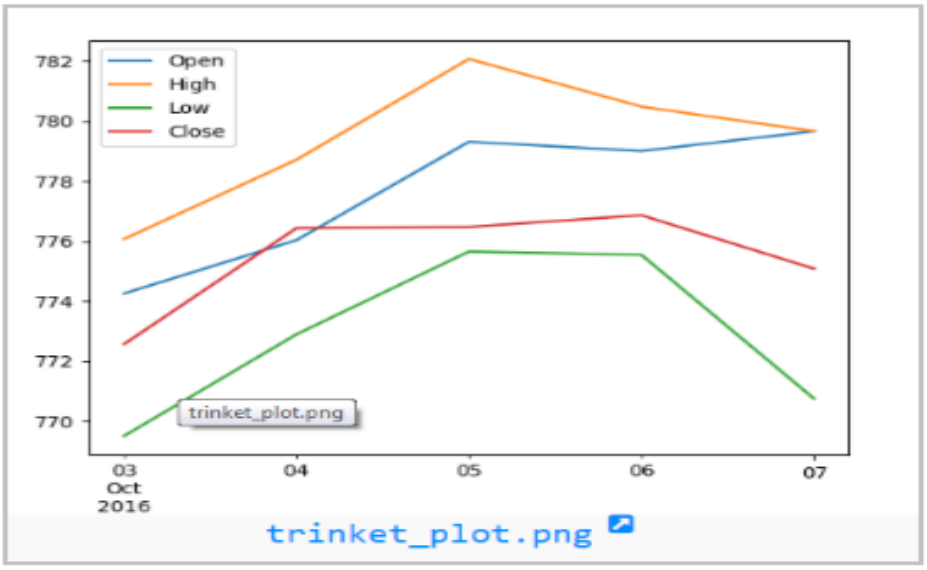
10-03-16,774.25,776.065002,769.5,772.559998

10-04-16,776.030029,778.710022,772.890015,776.429993

10-05-16,779.309998,782.070007,775.650024,776.469971

10-06-16,779,780.47998,775.539978,776.859985

10-07-16,779.659973,779.659973,770.75,775.080017



INPUT:

PROG24.PY - C:/Users/Supriya/OneDrive/Desktop/PROG24.PY (3.11.6)

File Edit Format Run Options Window Help

```
import matplotlib.pyplot as plt
from datetime import datetime
# Financial data
data = [
 {"Date": "10-03-16", "Open": 774.25, "High": 776.065002, "Low": 769.5, "Close": 772.559998},
 {"Date": "10-04-16", "Open": 776.030029, "High": 778.710022, "Low": 772.890015, "Close": 776.429993},
 {"Date": "10-05-16", "Open": 779.309998, "High": 782.070007, "Low": 775.650024, "Close": 776.469971},
 {"Date": "10-06-16", "Open": 779, "High": 780.47998, "Low": 775.539978, "Close": 776.859985},
 {"Date": "10-07-16", "Open": 779.659973, "High": 779.659973, "Low": 770.75, "Close": 775.080017}
]
# Extract dates and close prices
dates = [datetime.strptime(entry["Date"], "%m-%d-%y") for entry in data]
close_prices = [entry["Close"] for entry in data]
# Create a line chart
plt.figure(figsize=(10, 5))
plt.plot(dates, close_prices, marker='o', linestyle='-')
# Set labels and title
plt.xlabel('Date')
plt.ylabel('Close Price (USD)')
plt.title('Alphabet Inc. Financial Data (Oct 3, 2016 to Oct 7, 2016)')
# Format date on x-axis
plt.gcf().autofmt_xdate()
# Show the chart
plt.show()
```

Ln: 25 Col: 0

OUTPUT:

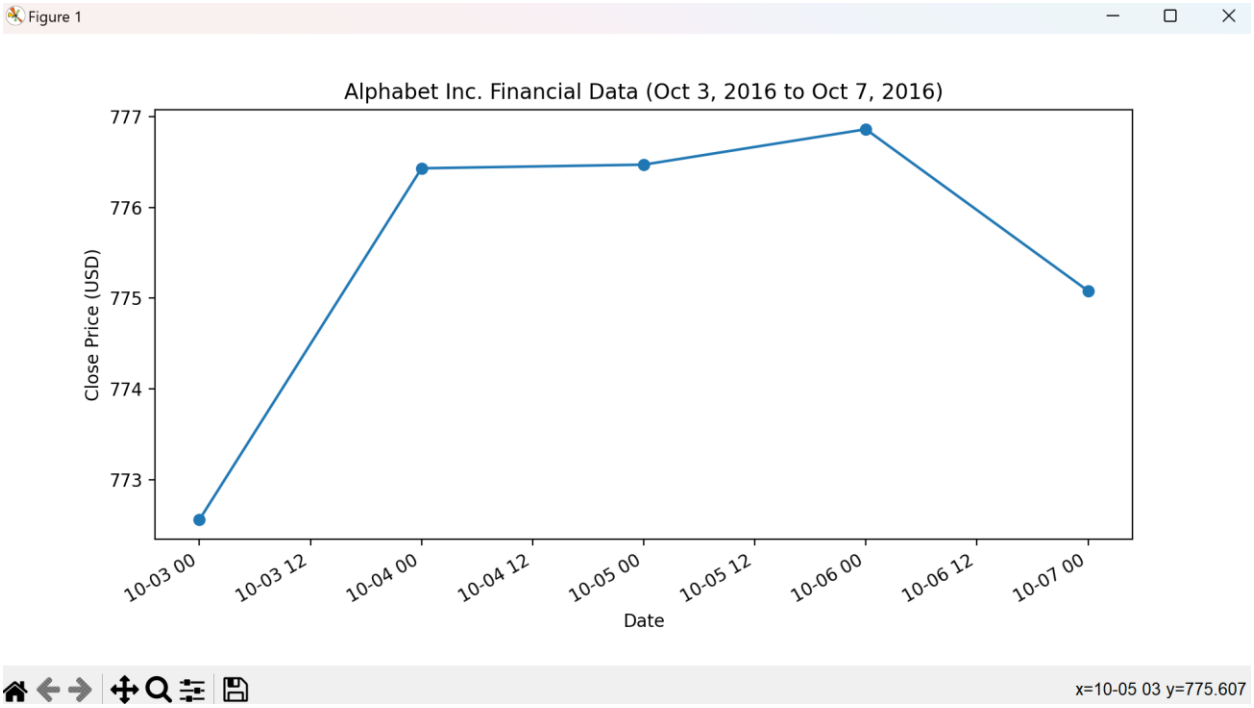


Figure 1
Alphabet Inc. Financial Data (Oct 3, 2016 to Oct 7, 2016)
Close Price (USD)
777
776
775
774
773
10-03 00
10-03 12
10-04 00
10-04 12
10-05 00
10-05 12
10-06 00
10-06 12
10-07 00
Date
x=10-05 03 y=775.607