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: import pandas as pd

```
pd.set_option('display.max_rows', None)
#pd.set_option('display.max_columns', None)
student_data = pd.DataFrame({
    'school_code': ['s001','s002','s003','s001','s002','s004'],
    'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'],
    'name': ['Alberto Franco','Gino Mcneill','Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],
    'date_Of_Birth ': ['15/05/2002','17/05/2002','16/02/1999','25/09/1998','11/05/2002','15/09/1997'],
    'age': [12, 12, 13, 13, 14, 12],
    ' height ': [173, 192, 186, 167, 151, 159],
    'weight': [35, 32, 33, 30, 31, 32],
    'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']},
    index=['S1', 'S2', 'S3', 'S4', 'S5', 'S6'])

print("Original DataFrame:")
print(student_data)
print('\nMean, min, and max value of age for each school with customized column names:')
grouped_single = student_data.groupby('school_code').agg(Age_Mean =
('age','mean'),Age_Max=('age',max),Age_Min=('age',min))
print(grouped_single)
```

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:

```
import pandas as pd

pd.set_option('display.max_rows', None)

#pd.set_option('display.max_columns', None)

student_data = pd.DataFrame({

    'school_code': ['s001','s002','s003','s001','s002','s004'],

    'class': ['V', 'V', 'VI', 'VI', 'V', 'VI'],

    'name': ['Alberto Franco','Gino Mcneill','Ryan Parkes', 'Eesha Hinton', 'Gino Mcneill', 'David Parkes'],

    'date_Of_Birth ': ['15/05/2002','17/05/2002','16/02/1999','25/09/1998','11/05/2002','15/09/1997'],

    'age': [12, 12, 13, 13, 14, 12],

    'height': [173, 192, 186, 167, 151, 159],

    'weight': [35, 32, 33, 30, 31, 32],
```

```
    'address': ['street1', 'street2', 'street3', 'street1', 'street2', 'street4']},
    index=['S1', 'S2', 'S3', 'S4', 'S5', 'S6'])

print("Original DataFrame:")
print(student_data)
print('\nSplit the said data on school_code wise:')
result = student_data.groupby(['school_code'])
for name,group in result:
    print("\nGroup:")
    print(name)
    print(group)
print("\nType of the object:")
print(type(result))
```

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]

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

Type of the object:
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>
```

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:

```
import pandas as pd

# Assuming your data is in a DataFrame called df

# If not, you should read the data into a DataFrame first

# Create a DataFrame (for example purposes)

data = {

   'Year': [1986, 1986, 1985, 1986, 1987],

   'WHO region': ['Western Pacific', 'Americas', 'Africa', 'Americas', 'Americas'],

   'Country': ['Viet Nam', 'Uruguay', "Cte d'Ivoire", 'Colombia', 'Saint Kitts and Nevis'],

   'Beverage Types': ['Wine', 'Other', 'Wine', 'Beer', 'Beer'],

   'Display Value': [0.00, 8.50, 1.62, 4.27, 1.98]

}

df = pd.DataFrame(data)

# Display the dimensions of the dataset

dimensions = df.shape

print("Dimensions of the dataset:", dimensions)

# Extract the column names

column_names = df.columns.tolist()

print("Column names:", column_names)
```

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:

```
import pandas as pd
df = pd.DataFrame({
   'name_code': ['c0001','1000c','b00c2', 'b2c02', 'c2222'],
   'date_of_birth ': ['12/05/2002','16/02/1999','25/09/1998','12/02/2022','15/09/1997'],
   'age': [18.5, 21.2, 22.5, 22, 23]
})
print("Original DataFrame:")
print(df)
print("\nIndex of a substring in a specified column of a dataframe:")
df['Index'] = list(map(lambda x: x.find('c', 0, 5), df['name_code']))
print(df)
```

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:

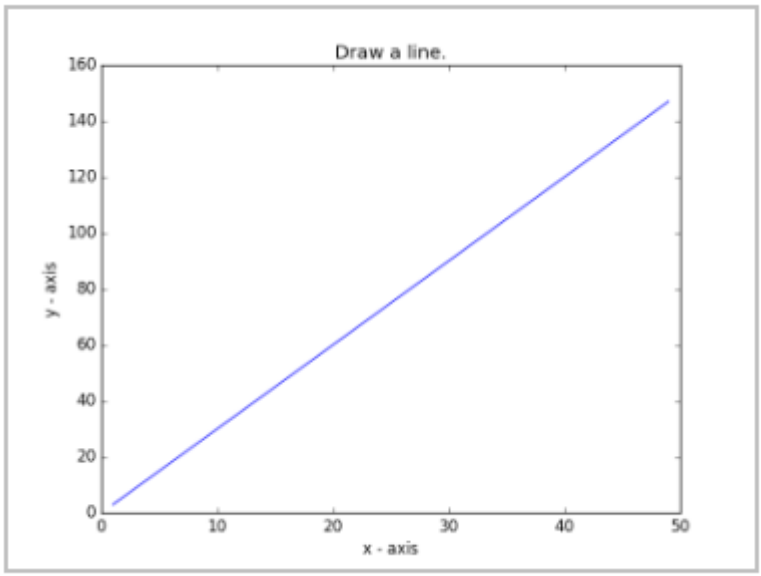
```
import pandas as pd
df = pd.DataFrame({
   'company_code': ['Abcd','EFGF', 'zefsalf', 'sdfslew', 'zekfsdf'],
   'date_of_sale': ['12/05/2002','16/02/1999','25/09/1998','12/02/2022','15/09/1997'],
```

```
    'sale_amount': [12348.5, 233331.2, 22.5, 2566552.0, 23.0]
})
print("Original DataFrame:")
print(df)
print("\nSwapp cases in comapny_code:")
df['swapped_company_code'] = list(map(lambda x: x.swapcase(), df['company_code']))
print(df)
```

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.

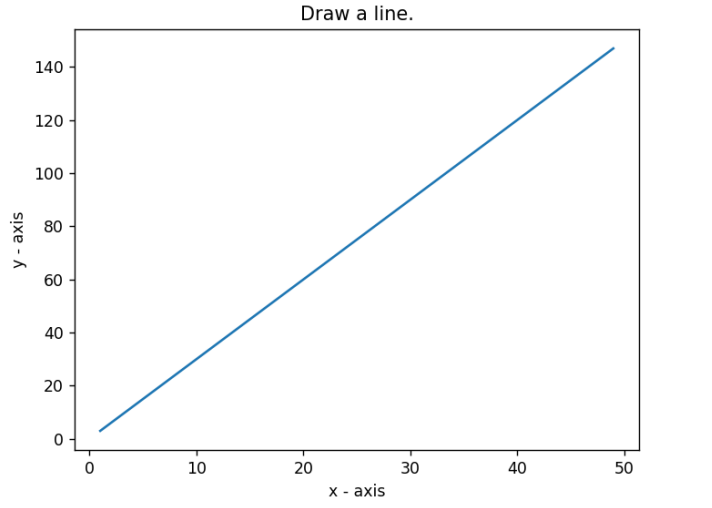


INPUT:

```
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()
```

OUTPUT:

```
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]
```



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.

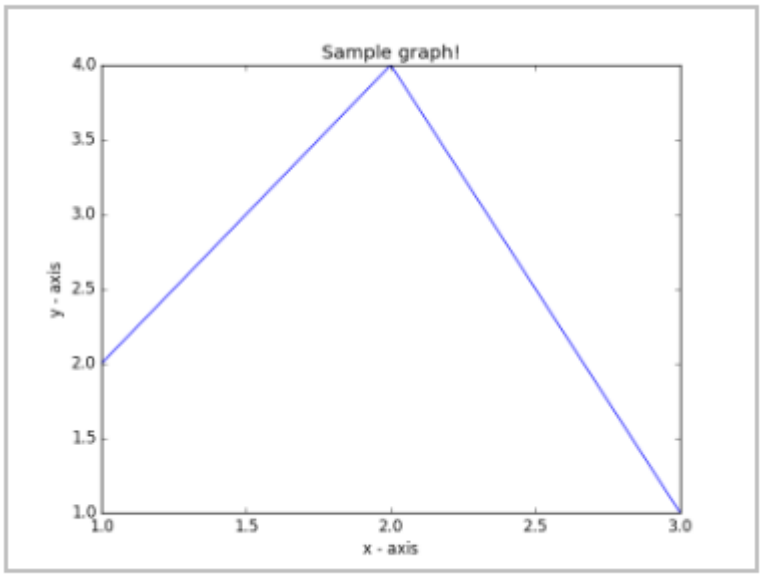
Test Data:

test.txt

1 2

2 4

3 1

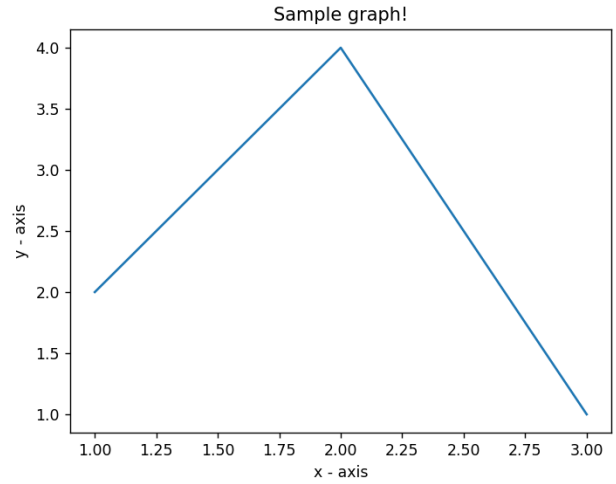


INPUT:

```
import matplotlib.pyplot as plt

# x axis values

x = [1,2,3]

# y axis values

y = [2,4,1]

# 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('Sample graph!')

# Display a figure.
```

plt.show()

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

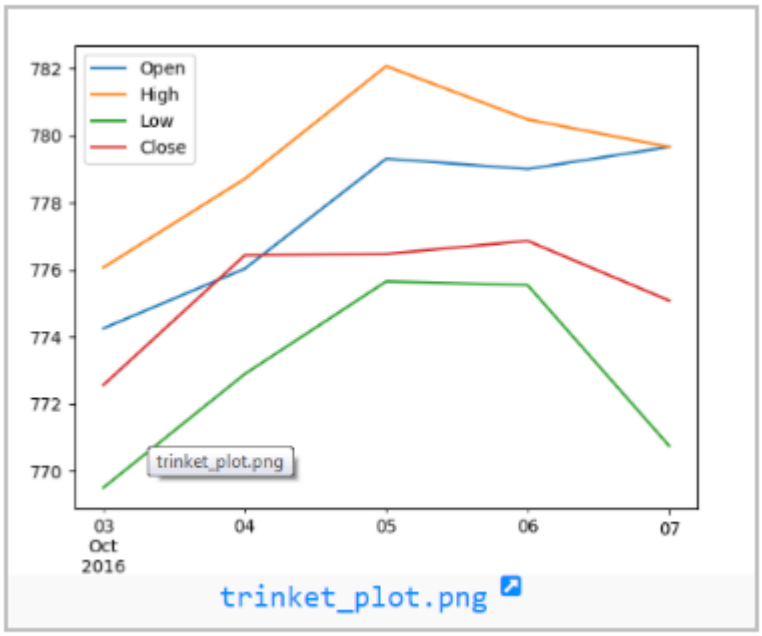
10-03-16,774.25,776.065002,769.5,772.559998

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:

```
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()
```

OUTPUT: