Data Visualization
Plotting with PyPlot

Data Visualization:

Data Visualization refers to graphical or visual representation of data and information using visual elements like charts, graphs and maps. Data Visualization immensely used for decision making because it unveils (shows) patterns, trends, relation and co-relation in the data.

Python matplotlib:

The python matplotlib is a high quality plotting library that provides many interface and functionality for 2D graphs.

PyPlot:

PyPlot is collection of methods in matplotlib library of Python which allows to construct 2-D plots easily.

Installation and importing PyPlot:

- Install the python interpreter in computer.
- Open the Command prompt and set your path
- C:\Users\Administrator\AppData\Local\Programs\Python\Python35-32\Scripts (This path will be different in your computer)
- Type following command and press enter key.

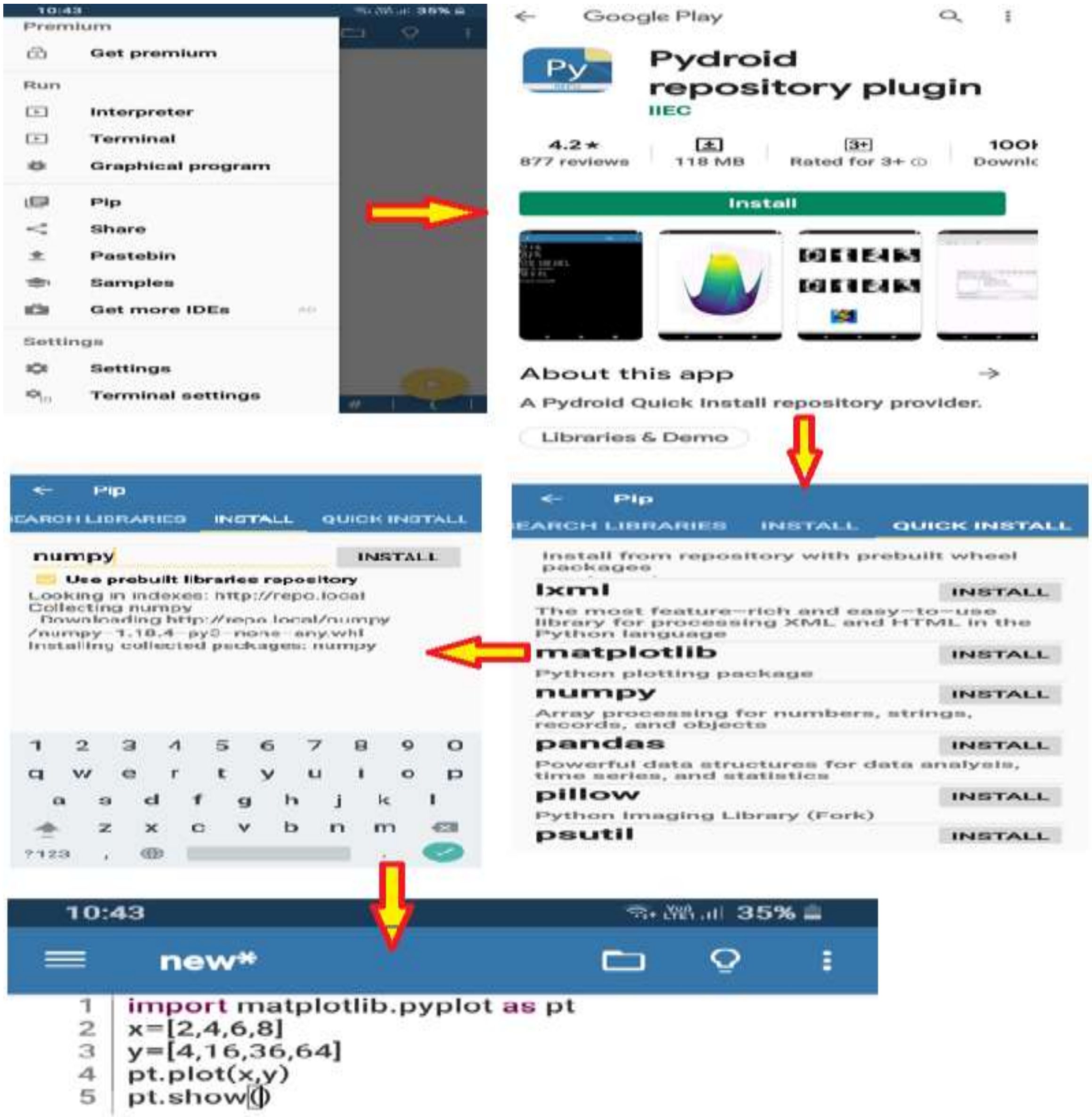
python –m pip install matplotlib

- Open Python IDLE environment and open new file for script mode and type following command to import the pyplot in IDLE environment.

  import matplotlib.pyplot   Or

  import matplotlib.pyplot   as plt

- Install the Library Andriod App as given below steps.



A Pydroid Quick Install repository provider.

numpy

Use prebuilt libraries repository

```
Looking in indexes: http://repo.local
Collecting numpy
  Downloading http://repo.local/numpy
/numpy-1.18.4-py3-none-any.whl
Installing collected packages: numpy
```

Install from repository with prebuilt wheel packages

lxml — INSTALL
The most feature-rich and easy-to-use library for processing XML and HTML in the Python language

matplotlib — INSTALL
Python plotting package

numpy — INSTALL
Array processing for numbers, strings, records, and objects

pandas — INSTALL
Powerful data structures for data analysis, time series, and statistics

pillow — INSTALL
Python Imaging Library (Fork)

psutil — INSTALL

```
import matplotlib.pyplot as pt
x=[2,4,6,8]
y=[4,16,36,64]
pt.plot(x,y)
pt.show()
```

Types of Plotting / Charts:

- **Line Chart:** It is type of chart / graph that display information as series of data points called marker. All markers are connected with straight line segments

  With pyplot, line chart can create by using plot( )

- **Bar Chart:** It shows the categories of data in form of rectangular bars with height or length. Bar chart may can be represent in horizontal or vertical forms.

  With pyplot, Vertical bar can create by using bar( )
  And Horizontal bar can create by using barh( )

- **Scatter Plot:** It is similar to line chart that except markers are not connected with line segments in this plot.

  With pyplot, Scatter plot can create by using scatter( )

- **Pie Chart:** It is a circular graph which is divided into slices to illustrate the numerical proportions.

  With pyplot, pie chart can create by using pie( )

- **Histogram Plot:** It represents the visual interpretation of numerical data by indicating the number of data points those lie between a ranges of values.

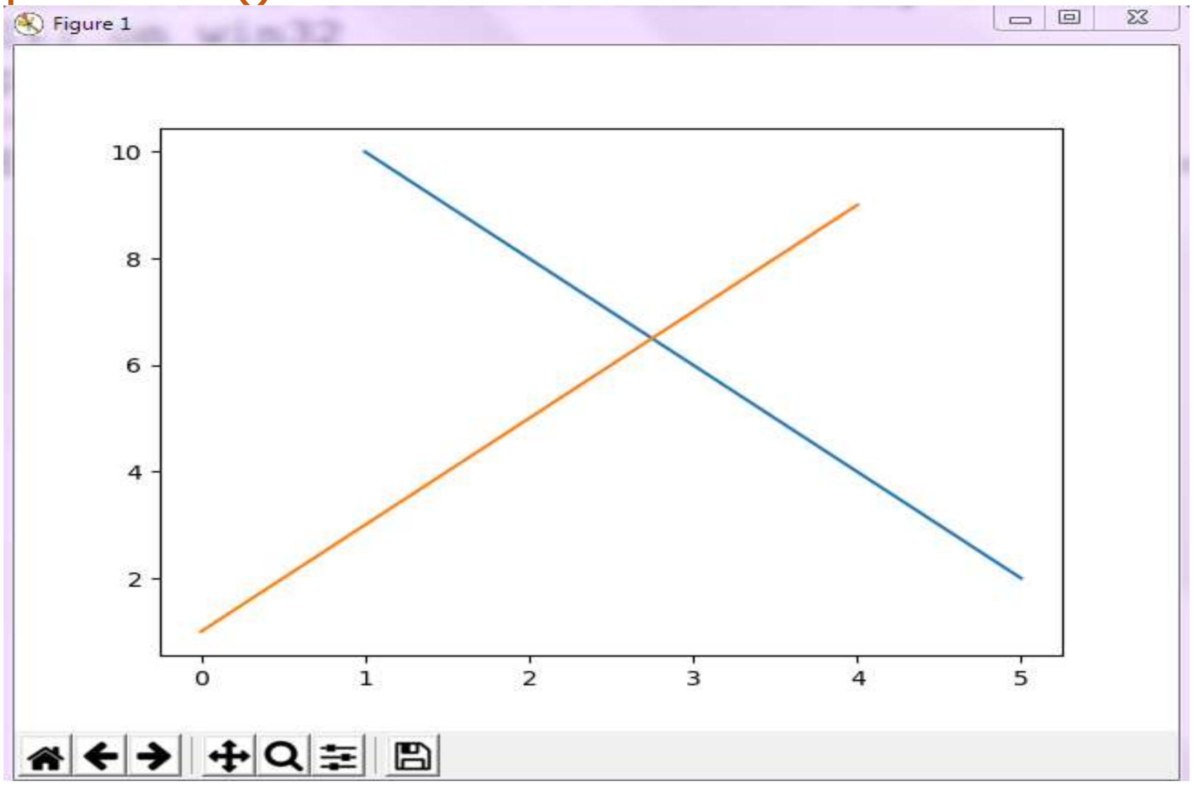
  With pyplot, Histogram plot can create by using hist( )

- **BoxPlot chart:** A box plot is the visual representation of statistical five numbers of summary of given data set.

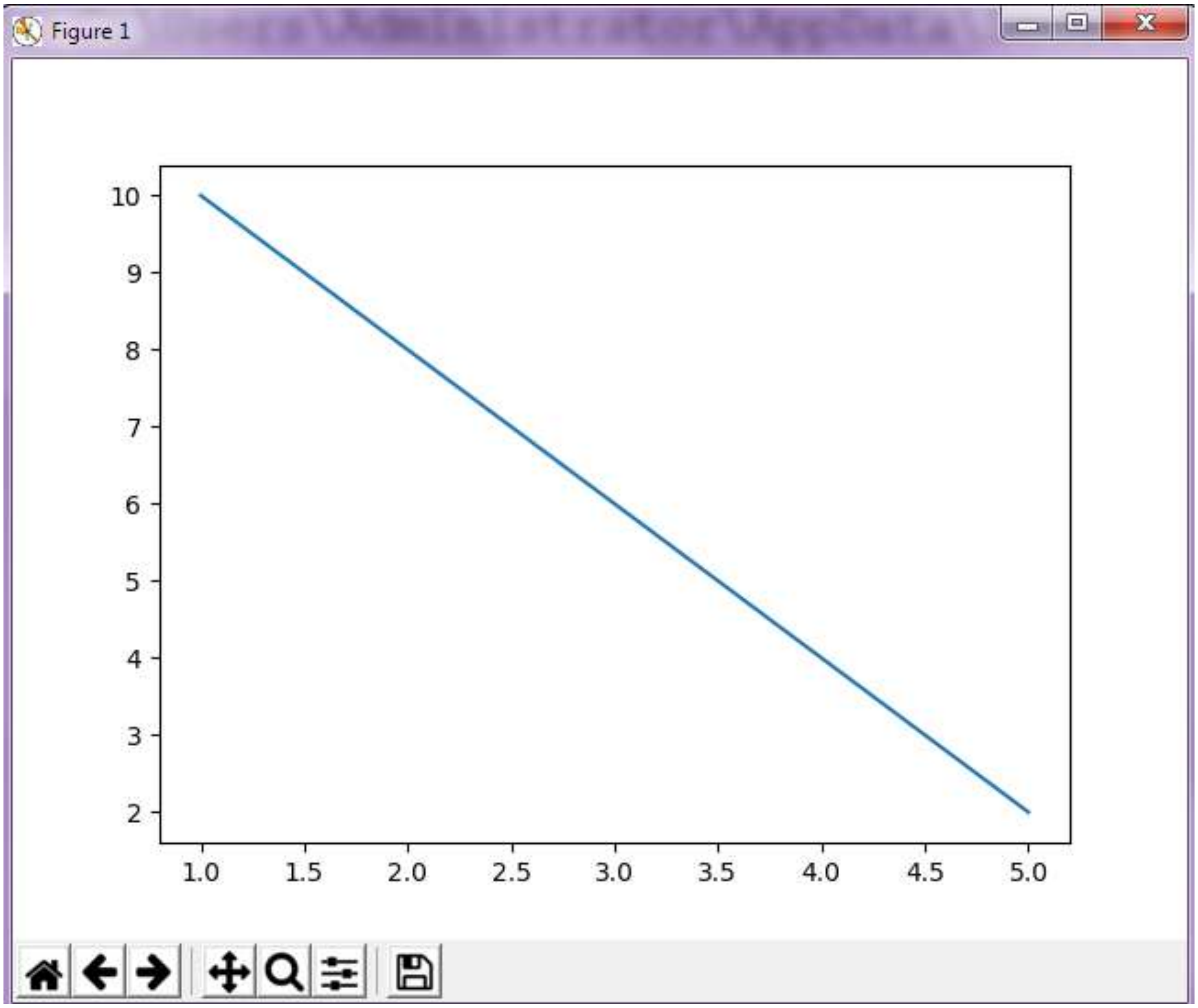
  With pyplot, boxplot can create by using boxplot( )

Creation of Line graph

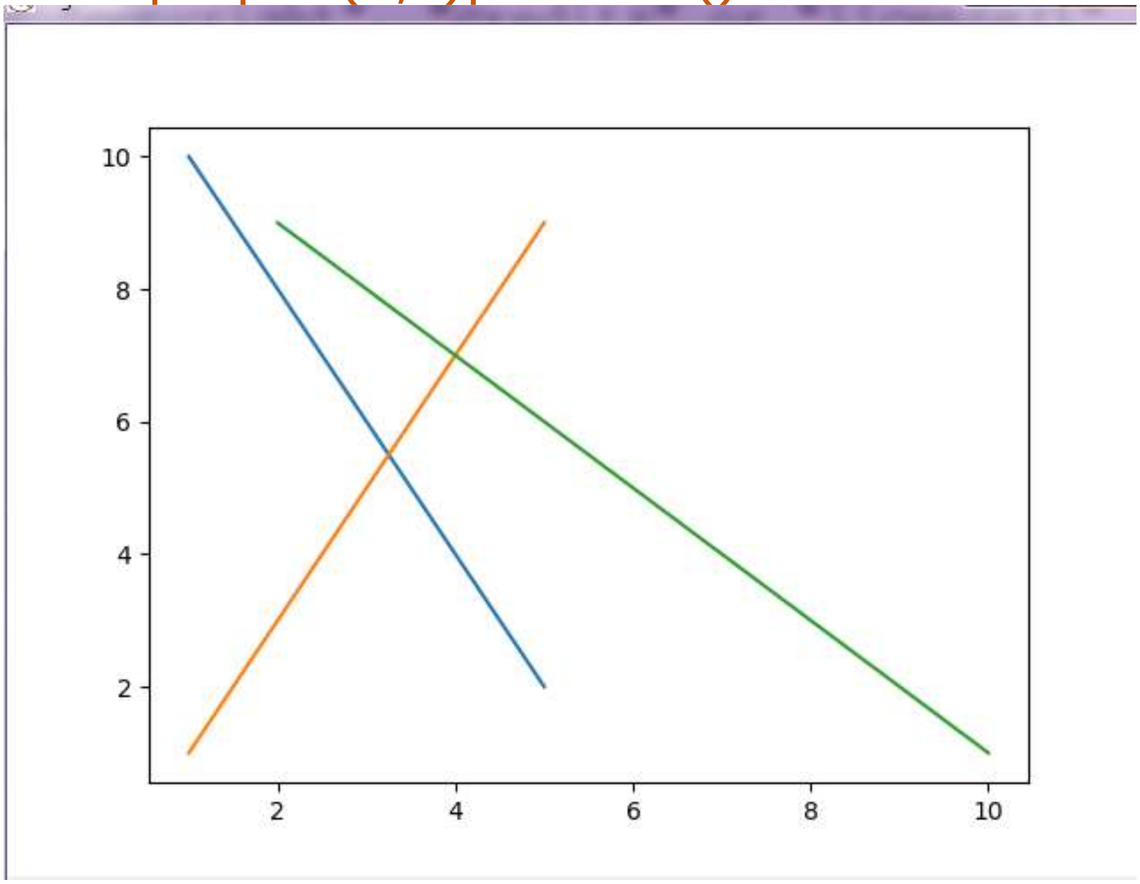
```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[10,8,6,4,2]
c=[1,3,5,7,9]
plt.plot(a,b,c)
plt.show()
```



```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[10,8,6,4,2]
c=[1,3,5,7,9]
plt.plot(a,b)
plt.show()
```



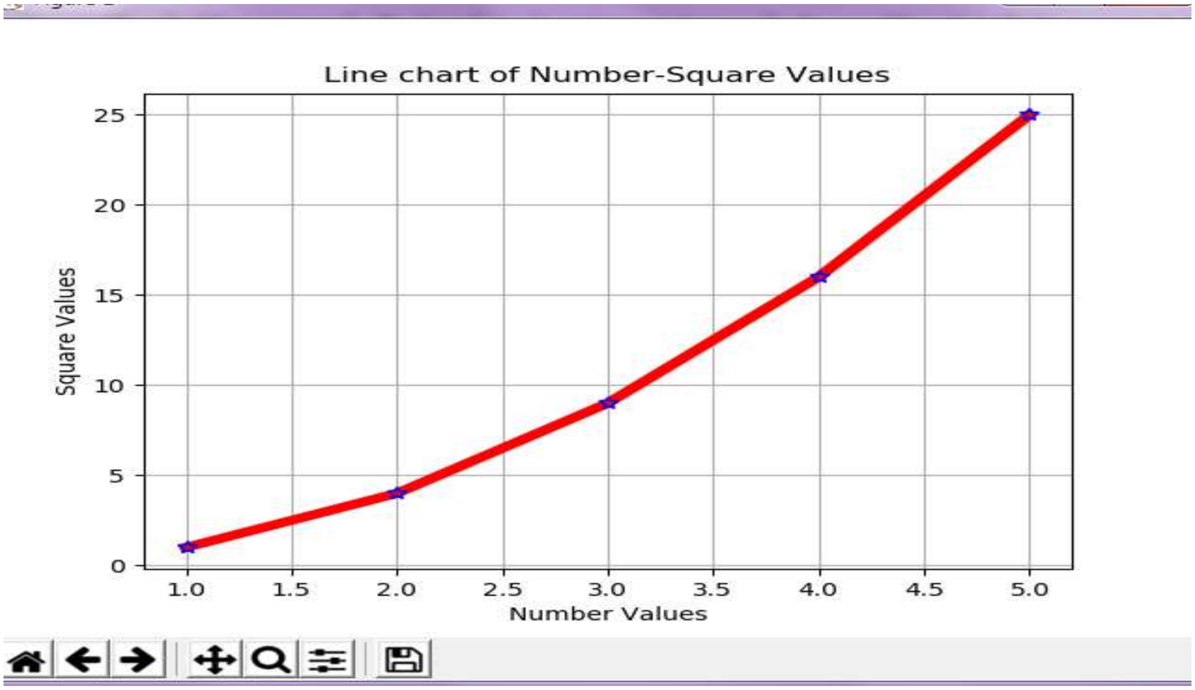
```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[10,8,6,4,2]
c=[1,3,5,7,9]
plt.plot(a,b)
plt.plot(a,c)
plt.plot(b,c)plt.show()
```



Basic color codes of Matplotlib

Character	Color	Character	Color	Character	Color
‘b’	Blue	‘m’	Magenta	‘c’	Cyan
‘g’	Green	‘y’	Yellow	‘w’	White
‘r’	Red	‘k’	Black		

```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[1,4,9,16,25]
plt.plot(a,b,color='r', linewidth=5, markersize=7,
        markeredgecolor='b', marker='*')
plt.xlabel("Number Values")
plt.ylabel("Square Values")
plt.title("Line chart of Number-Square Values")
plt.grid(True)
plt.show()
```



Example:

First 10 terms of Fibonacci series are stored in a list.

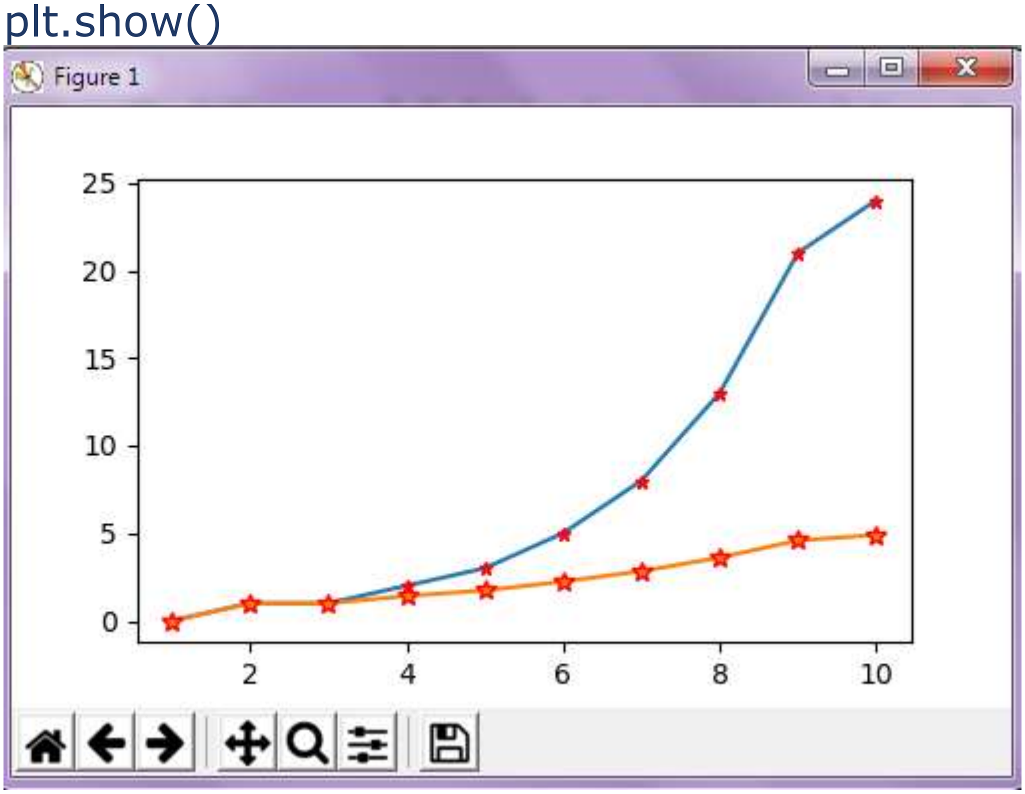
Fib=[0,1,1,2,3,5,8,13,21,24]

Write a program to plot Fibonacci terms and their square roots with two separate line on the same plot.

- First Fibonacci series should be plotted as a cyan line with 'o' marker having size as 5 and edge color as red
- The square root series should be plotted as a black line with '+' marker having size as 7 and edge color as red

Solution:

```
import matplotlib.pyplot as plt
import numpy as np
fib=[0,1,1,2,3,5,8,13,21,24]
sqfib=np.sqrt(fib)
plt.figure(figsize=(5,3))
plt.plot(range(1,11), fib, markersize=5, linestyle='solid',
        markeredgecolor='red', marker='*')

plt.plot(range(1,11), sqfib, markersize=7,
linestyle='solid', markeredgecolor='red', marker='*')
plt.show()
```



Example:

The given reversed Fibonacci series elements stored in a list.

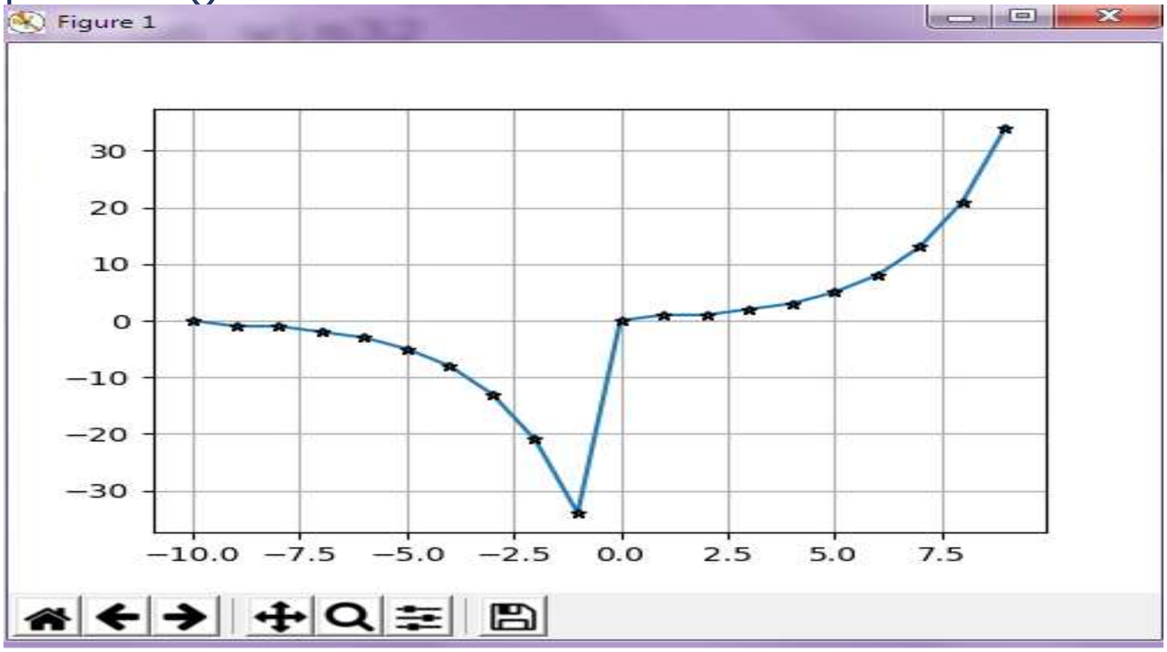
nfib=[0,-1,-1,-2,-3,-5,-8,-13,-21,-34,0,1,1,2,3,5,8,13,21,34]

Write a program to plot nfib with following specification.

- The line color should be magenta
- The marker edge color should be black with size 5
- Grid should be displayed

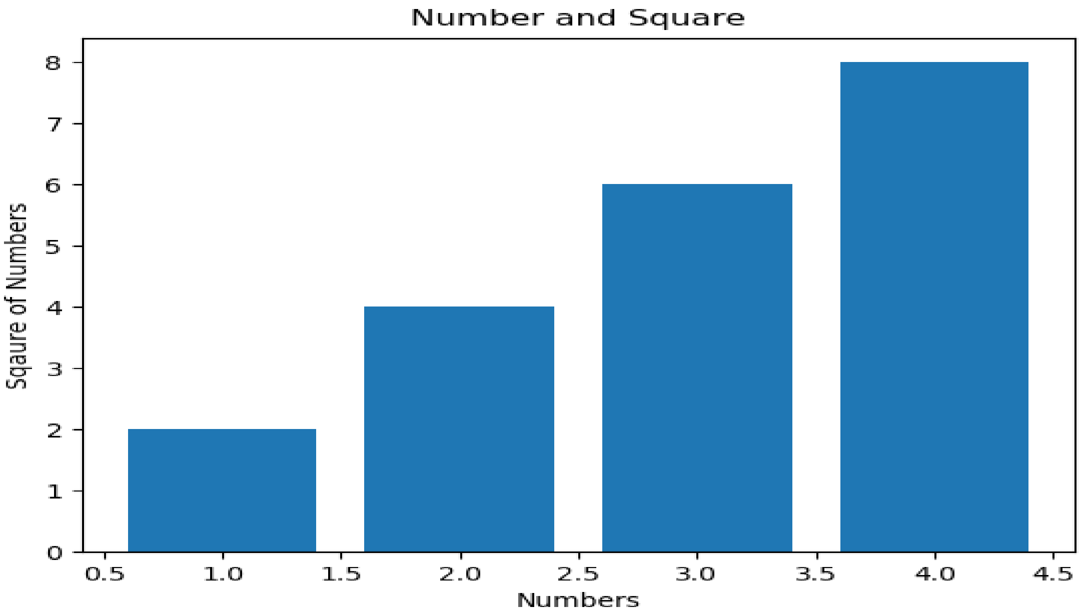
Solution:

```
import matplotlib.pyplot as plt

import numpy as np

nfib=[0,-1,-1,-2,-3,-5,-8,-13,-21,-34,0,1,1,2,3,5,8,13,21,34]

plt.figure(figsize=(5,4))

plt.plot(range(-10,10),nfib,markeredgecolor='k',
markersize=5,

        linestyle='solid', marker='*')

plt.grid(True)
plt.show()
```

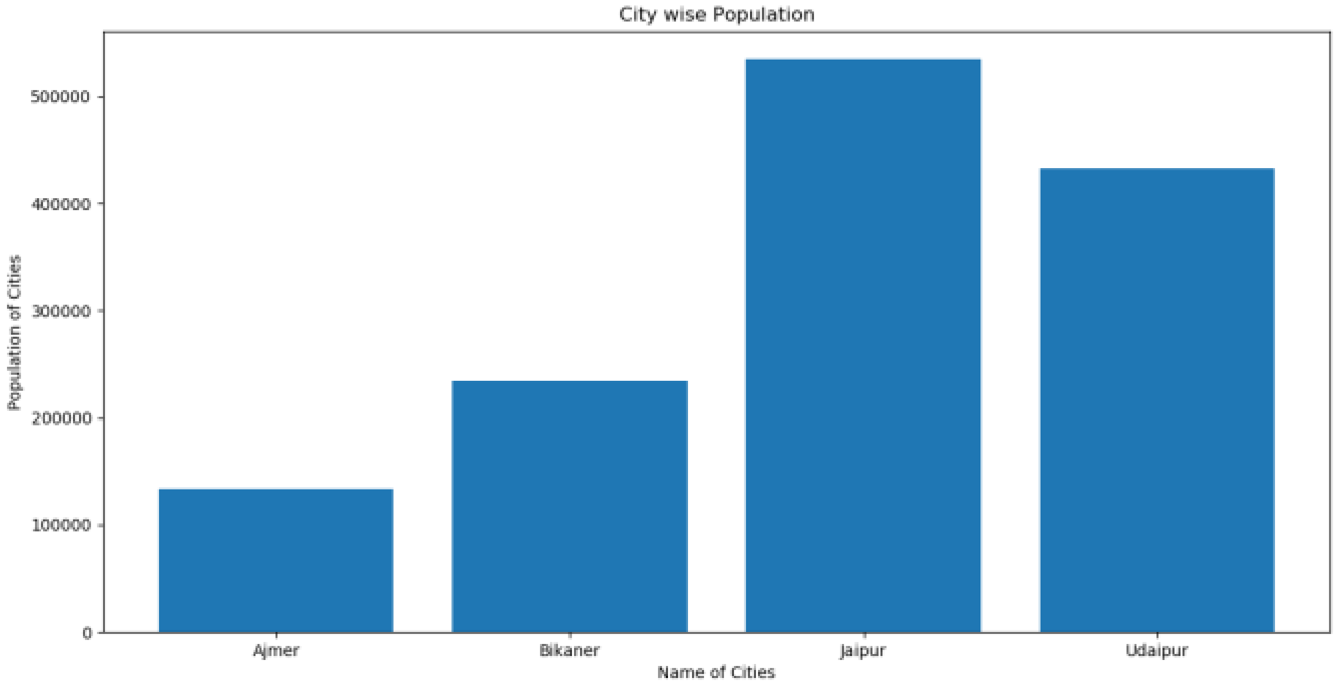


Creation of bar graph

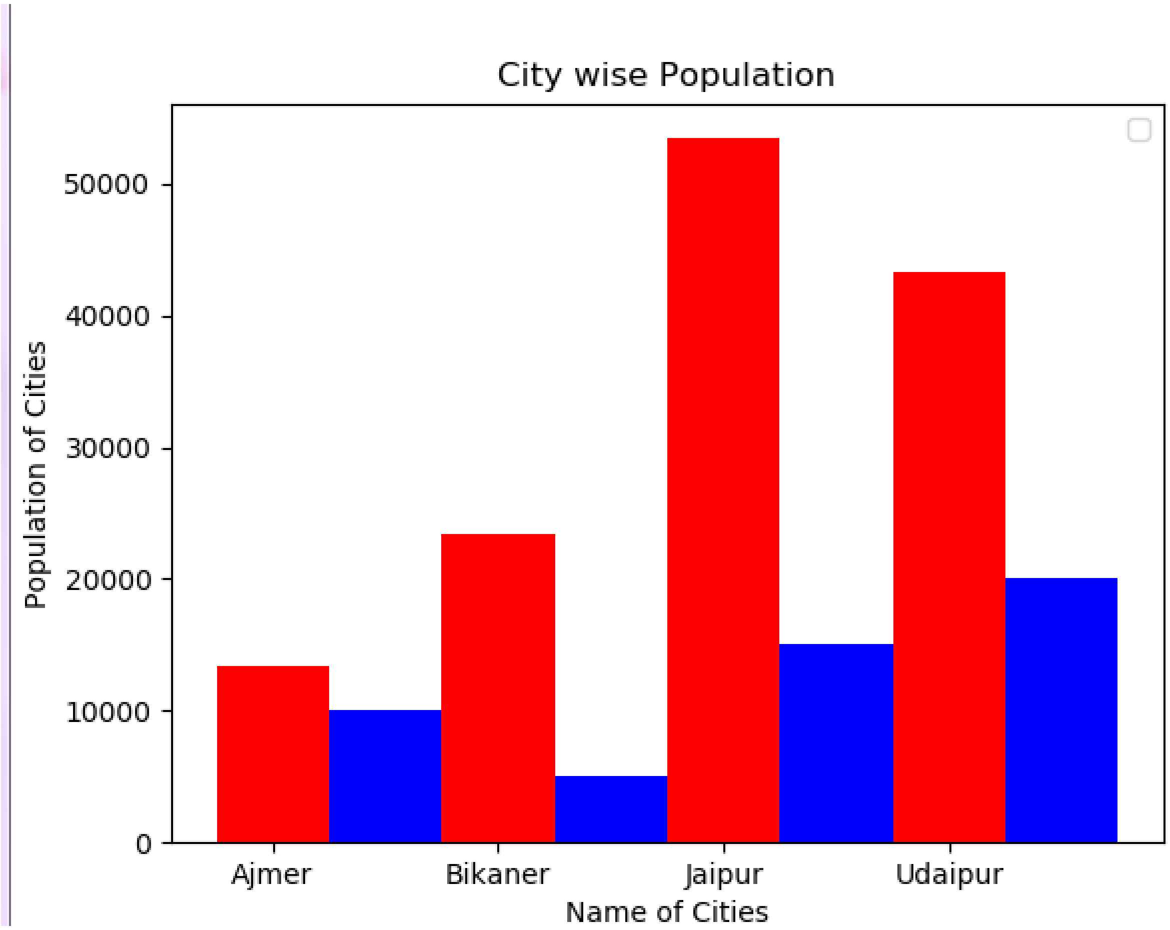
```
import matplotlib.pyplot as plt
a=[1,2,3,4]
b=[2,4,6,8]
c=[1,4,9,16]
plt.bar(a,b)
plt.xlabel("Numbers")
plt.ylabel("Sqaure of Numbers")
plt.title("Number and Square")
plt.show()
```



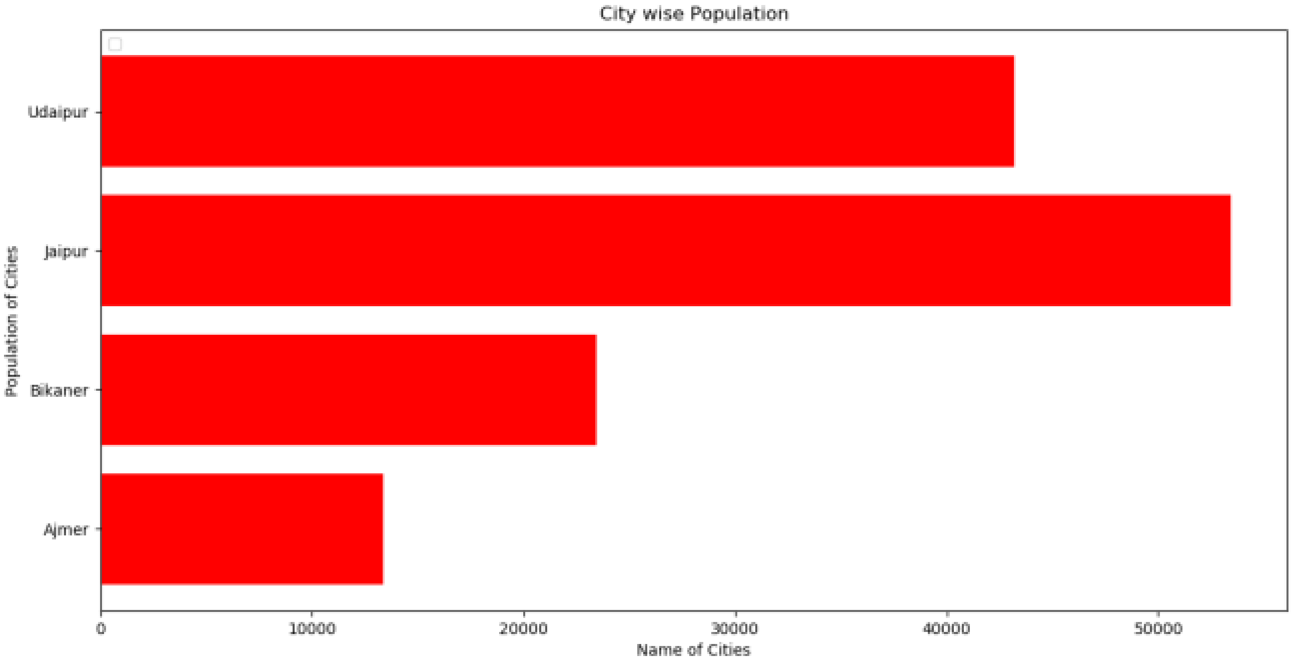
```
cities=['Ajmer','Bikaner','Jaipur','Udaipur']
population=[133454,234525,534223,432123]
plt.bar(cities,population)
plt.xlabel("Name of Cities")
plt.ylabel("Population of Cities")
plt.title("City wise Population")
plt.show()
```



```
import matplotlib.pyplot as plt

import numpy as np

plt.xlabel("Name of Cities")

plt.ylabel("Population of Cities")

plt.title("City wise Population")

cities=['Ajmer','Bikaner','Jaipur','Udaipur']

density=[10000,5000,15000,20000]

population=[13345,23452,53422,43212]

x=np.arange(4)

plt.bar(cities,population,
width=0.5,color='red')#,color=['r','g','c','b'])

plt.bar(x+0.5,density, width=0.5,color='blue')#,color=['g','r','b','c'])

plt.show()
```



```
import matplotlib.pyplot as plt
plt.xlabel("Name of Cities")
plt.ylabel("Population of Cities")
plt.title("City wise Population")
cities=['Ajmer','Bikaner','Jaipur','Udaipur']
population=[13345,23452,53422,43212]
plt.barh(cities,population,color='red')
plt.legend(loc='upper left')
plt.show()
```


City wise Population
Udaipur
Jaipur
Bikaner
Ajmer
Population of Cities
0
10000
20000
30000
40000
50000
Name of Cities

Finish