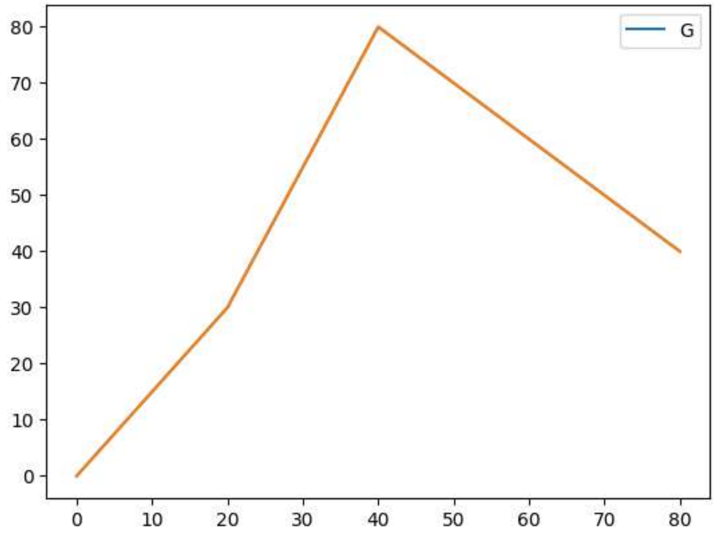
Name : Sakshi Ravindra Aher

PRN : 202201040089

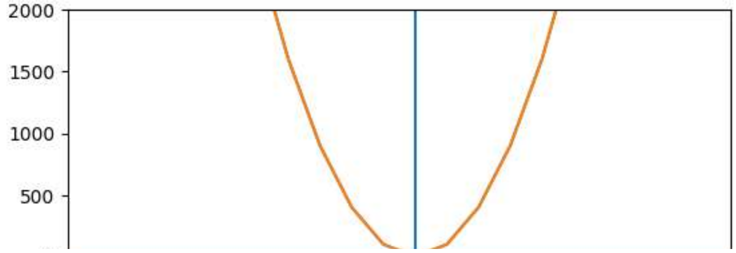
Roll no : 602

Div : F(f-1)

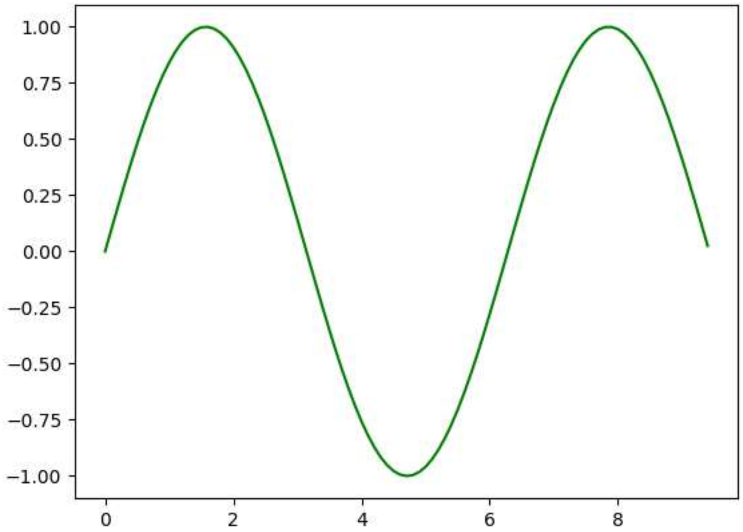
```
import matplotlib.pyplot as plt
#create data for plotting
X_values =[0, 20, 40, 60, 80 ]
Y_values = [0, 30,80, 60, 40 ]
plt.plot(X_values,Y_values)
plt.Ylabel = ("X_axis")
plt.Xlabel = ("Y_axis")
plt.legend("Graph")
plt.plot(X_values,Y_values)
plt.show()
```



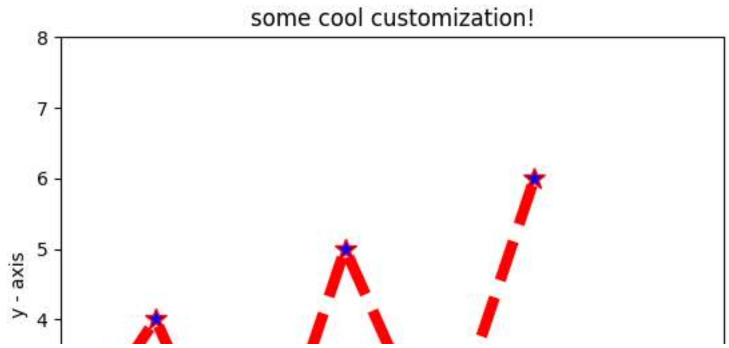
```
Y1=[]
Y2=[]
X=range(-100,100,10)
for i in X: Y1.append(i**2)
for i in X: Y2.append(i**2)
plt.plot(X,Y1)
plt.plot(X,Y2)
plt.xlabel("X")
plt.ylabel("Y")
plt.ylim(-2000,2000)
plt.axhline(0)
plt.axvline(0)
plt.savefig("quad.png")
plt.show()
```



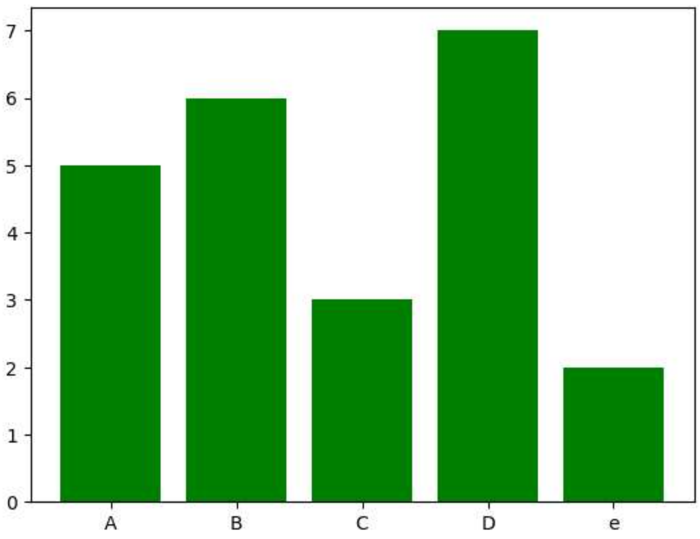
```
import numpy as np
x=np.arange(0,3*np.pi,0.1)
y1=np.sin(x)
plt.plot(x,y1,'green')
plt.show()
```



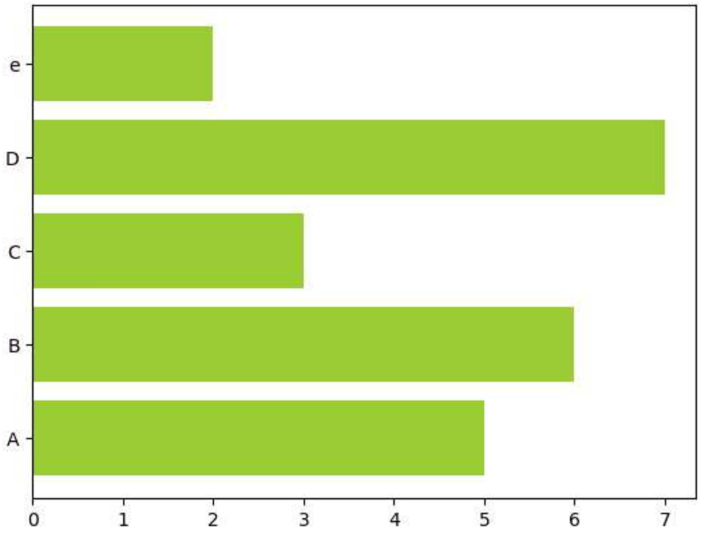
```
import matplotlib.pyplot as plt
# x axis values
x=[1,2,3,4,5,6]
# y axis values
y=[2,4,1,5,2,6]

#plotting the points
plt.plot(x, y, color='r', linestyle='dashed', linewidth=6, marker='*', markerfacecolor='blue', markersize=12)

# setting x and y axis range
plt.ylim(1,8)
plt.xlim(1,8)

#naming the  x axis
plt.xlabel('x - axis')

#naming the  x axis
plt.ylabel('y - axis')

#giving a title to my graph
plt.title('some cool customization!')

# function to show the plot
plt.show()
```



```
import matplotlib.pyplot as plt

#create data for plotting
values= [5, 6, 3, 7, 2]
names= ["A","B","C","D","e"]
plt.bar(names, values, color ='g')
plt.show()
```

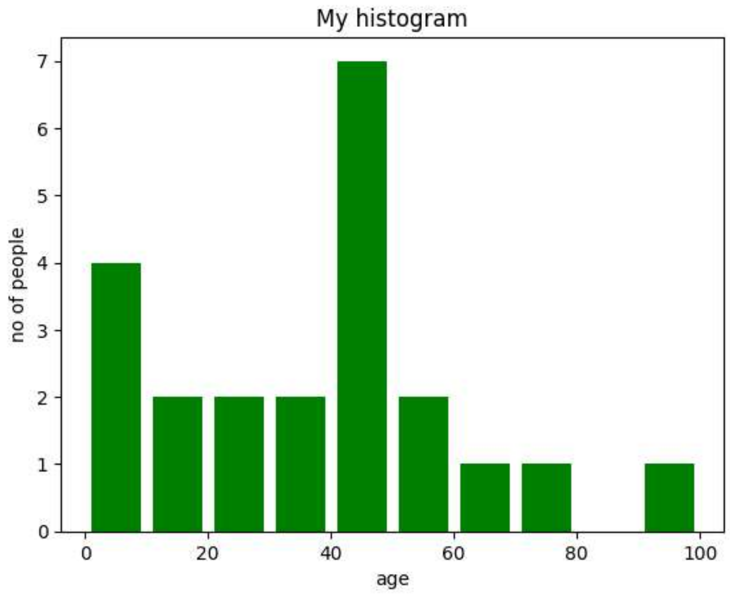


```
plt.barh(names, values, color="yellowgreen")
plt.show()
```

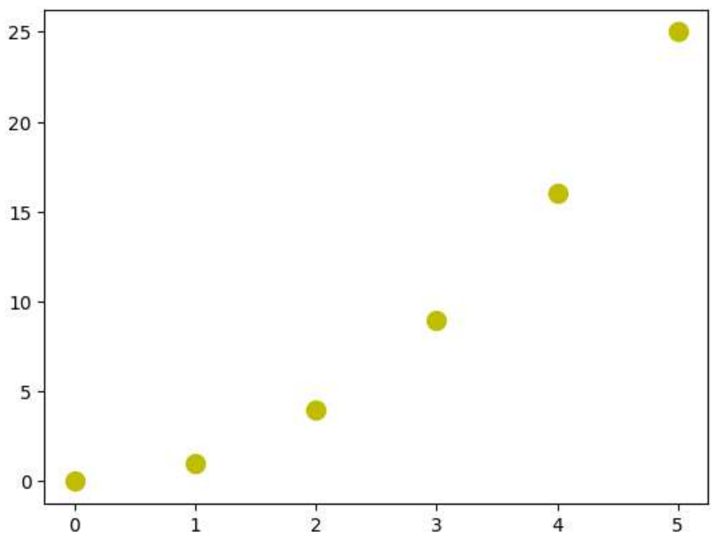


```
import matplotlib.pyplot as plt

# frequencies
```

```
ages=[2, 5, 70, 40, 30, 45, 50, 45, 43, 40, 44, 60, 7, 13, 57, 18, 90, 7, 32, 21, 20, 40]

#setting the ranges and no of intervals
range =(0,100)
bins = 10

#plotting a histogram
plt.hist(ages, bins, range, color='green', histtype='bar',rwidth=0.8)
#x-axis label
plt.xlabel('age')
#frequency label
plt.ylabel('no of people')
#plot title
plt.title('My histogram')

plt.show()
```

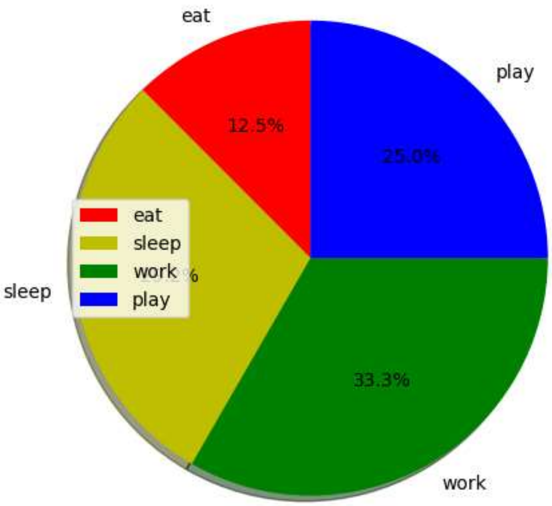


```
x_values=[0,1,2,3,4,5]
y_values=[0,1,4,9,16,25]
plt.scatter(x_values, y_values, s=100, color='y')
plt.show()
```



```
#pie chart
import matplotlib.pyplot as plt

#defining labels
activities=['eat','sleep','work','play']
```

```
#portion covered by each label
slices=[3,7,8,6]

#color for each label
colors=['r','y','g','b']

#plotting the pie chart
plt.pie(slices,labels = activities, colors=colors, startangle=90, shadow= True, explode=(0,0,0,0),radius=1.2, autopct='%1.1f%%')

#plotting legend
plt.legend()

#showing the plot
plt.show()
```



```
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd

data_set=pd.read_csv('/content/sample_data/Salary_Data (3).xls')
data_set
```

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

```
x=data_set.iloc[:,:-1].values
y=data_set.iloc[:,1].values
print(x)
print(y)
```

```
[[ 1.1]
 [ 1.3]
 [ 1.5]
 [ 2. ]
 [ 2.2]
 [ 2.9]
 [ 3. ]
 [ 3.2]
 [ 3.2]
 [ 3.7]
 [ 3.9]
 [ 4. ]
 [ 4. ]
 [ 4.1]
 [ 4.5]
 [ 4.9]
 [ 5.1]
 [ 5.3]
 [ 5.9]
 [ 6. ]
 [ 6.8]
 [ 7.1]
 [ 7.9]
 [ 8.2]
 [ 8.7]
 [ 9. ]
 [ 9.5]
 [ 9.6]
 [10.3]
 [10.5]]
[ 39343.  46205.  37731.  43525.  39891.  56642.  60150.  54445.  64445.
  57189.  63218.  55794.  56957.  57081.  61111.  67938.  66029.  83088.
  81363.  93940.  91738.  98273. 101302. 113812. 109431. 105582. 116969.
 112635. 122391. 121872.]
```

```
#splitting the dataset into traning and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test=train_test_split(x, y, test_size= 1/3, random_state=0)
```

```
print(x_train)
```

```
[[ 2.9]
 [ 5.1]
 [ 3.2]
 [ 4.5]
 [ 8.2]
 [ 6.8]
 [ 1.3]
 [10.5]
 [ 3. ]
 [ 2.2]
 [ 5.9]
 [ 6. ]
 [ 3.7]
 [ 3.2]
 [ 9. ]
 [ 2. ]
 [ 1.1]
 [ 7.1]
 [ 4.9]
 [ 4. ]]
```

```
print(y_train)
```

```
[ 56642.  66029.  64445.  61111. 113812.  91738.  46205. 121872.  60150.
  39891.  81363.  93940.  57189.  54445. 105582.  43525.  39343.  98273.
  67938.  56957.]
```

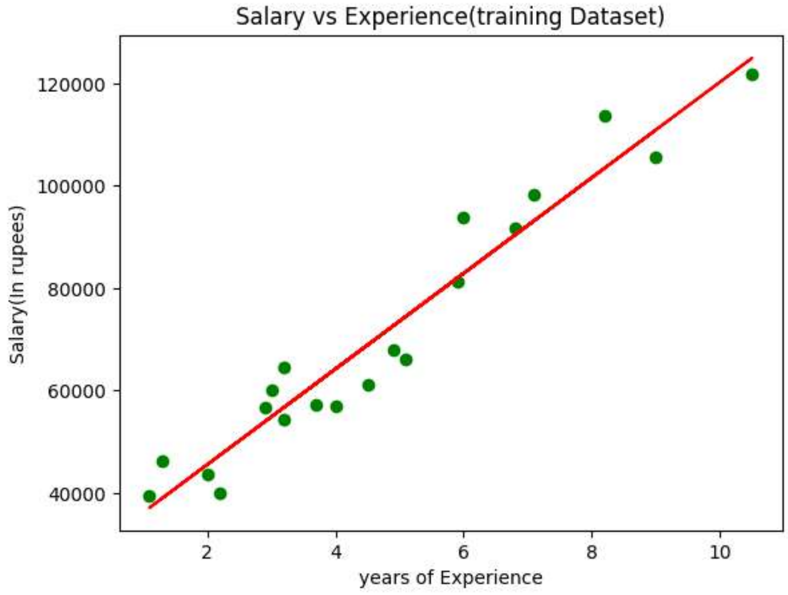
```
#fitting the simple linear Regression model to traning dataset
from sklearn.linear_model import LinearRegression
regressor= LinearRegression()
regressor.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

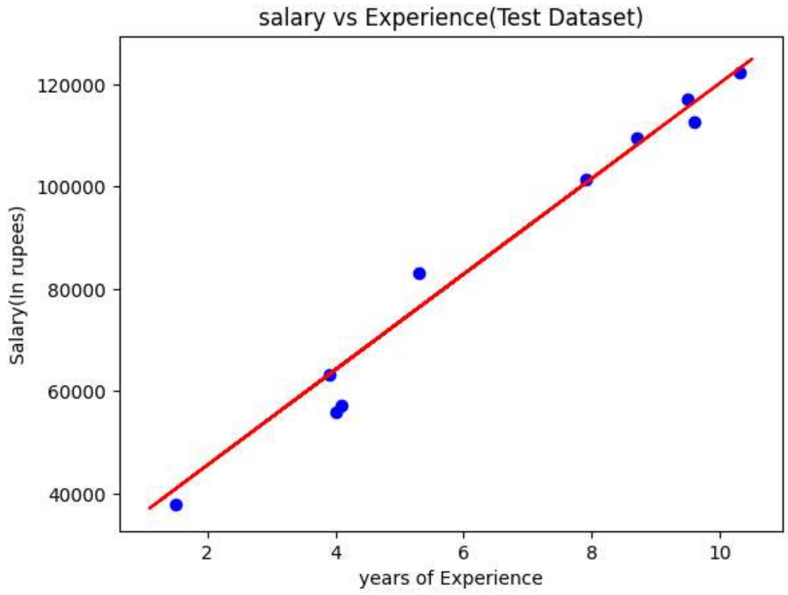
```
#prediction of test and training set result
y_pred= regressor.predict(x_test)
x_pred= regressor.predict(x_train)
print(x_pred)
print(y_pred)
```

```
[ 53919.42532909  74480.49870396  56723.20806202  68872.93323808
 103452.92027763  90368.60085726  38965.91742009 124948.58789682
  54854.0195734   47377.2656189   81957.25265845  82891.84690277
  61396.17928358  56723.20806202 110929.67423213  45508.07713028
  37096.72893147  93172.3835902   72611.31021533  64199.96201652]
 [ 40835.10590871 123079.39940819  65134.55626083  63265.36777221
  115602.64545369 108125.8914992  116537.23969801  64199.96201652
   76349.68719258 100649.1375447 ]
```

```
mtp.scatter(x_train, y_train, color='green')
mtp.plot(x_train, x_pred, color='red')
mtp.title("Salary vs Experience(training Dataset)")
mtp.xlabel("years of Experience")
mtp.ylabel("Salary(In rupees)")
mtp.show()
```



```
#visualizing the test set result
mtp.scatter(x_test,y_test,color="blue")
mtp.plot(x_train,x_pred,color="red")
mtp.title("salary vs Experience(Test Dataset)")
mtp.xlabel("years of Experience")
mtp.ylabel("Salary(In rupees)")
mtp.show()
```



● ×