

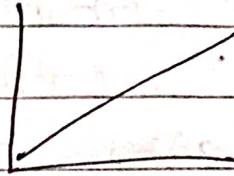
## Fourth program.

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = [10, 20, 30]
```

```
y = [5, 10, 15]
```

```
plt.plot(x, y)
```



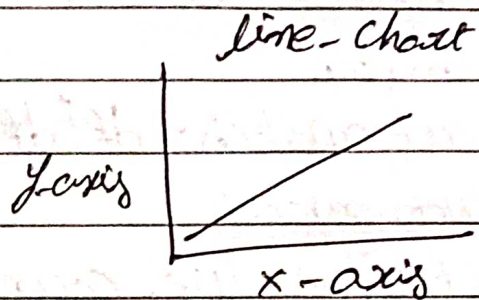
it display line chart.

```
plt.plot(x, y)
```

```
plt.xlabel('x-axis')
```

```
plt.ylabel('y-axis')
```

```
plt.title('line-chart')
```



it display line chart with title and label names

```
plt.plot(x, y, marker='o', mec='black', mfc='green',
         ms='20', ls='Solid')
```

ls  $\rightarrow$  line style default solid, 'dotted', 'dashed', 'dashdot'

marker  $\rightarrow$  o  $\rightarrow$  represents circle.

'x'  $\rightarrow$  Star

's'  $\rightarrow$  Square

1  $\rightarrow$  Tick Down

'.'  $\rightarrow$  Point

'D'  $\rightarrow$  Diamond

2  $\rightarrow$  Tick Up

'+'  $\rightarrow$  Plus

'P'  $\rightarrow$  Pentagon

3  $\rightarrow$  Left

'p'  $\rightarrow$  Plus

'H'  $\rightarrow$  Hexagon

4  $\rightarrow$  Right

ms  $\rightarrow$  To increase or decrease marker size.

mec  $\rightarrow$  marker external color

mfc  $\rightarrow$  marker face color

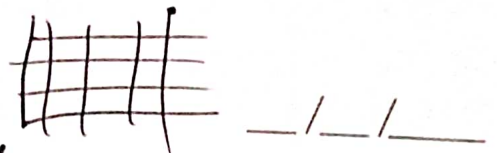
# Grid

```
x = [1, 2, 3, 4, 5]
```

```
y = [30, 20, 10, 40, 50]
```

```
plt.plot(x, y, marker='o', ms='20', ls='Solid', mfc='red',
         mec='green')
```





`plt.grid(axis="both", ls='solid', color='green', lw=1)`

`lw` → line width

`plt.grid(axis='x')` → display x-axis grid lines

## # Legend

`import matplotlib.pyplot as plt`

`import numpy as np`

`x = np.arange(0, 5, 1)`

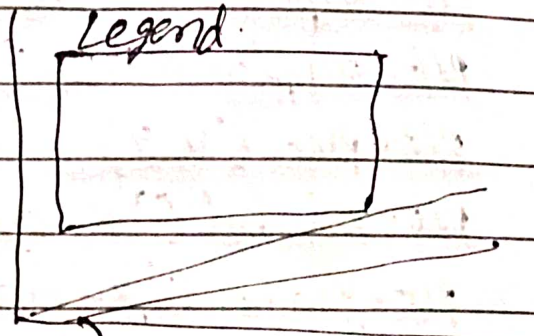
`y = x**2`

`y1 = x**3`

`plt.plot(x, y, marker='o', label='Squares')`

`plt.plot(x, y1, marker='x', label='cubes')`

`plt.legend(title='Charts', loc=2, framealpha=1,  
facecolor='yellow', edgecolor='black', fancybox=True,  
shadow=True)`



`for index, value in enumerate(y):`

`plt.text(index, value, str(value))`

`for index, value in enumerate(y1):`

`plt.text(index, value, str(value))`

`arange(0, 5, 1)` → create array of elements  
from starting 0 to 4 spacing b/w values 1

`plt.text(index, value, str(value))`

↓  
x-coordinate

↓  
y-coordinate

→ Convert the value to string  
for display as text.



## # Subplots

x = [1, 2, 3, 4, 5]

y = [20, 10, 30, 40, 50]

y1 = [5, 8, 10, 15, 20]

plt.suptitle("Whole Figure")

plt.subplot(1, 2, 1)

plt.plot(x, y)

plt.xlabel('x-axis')

plt.ylabel('y-axis')

plt.title('First plot')

plt.subplot(1, 2, 2)

plt.plot(x, y1)

plt.title('Second plot')

## # Scatter

import random

x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

y = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

random.shuffle(y)

plt.scatter(x, y, color='green', marker='x', s=15,  
linewidth=10, alpha=1)

x1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

y1 = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

plt.scatter(x1, y1, color='purple', marker='s', s=100)



\_/\_/\_

```
# Scatter calcd with each point
```

```
x=[10, 20, 30, 40, 50]
```

```
y=[2, 10, 40, 50, 100]
```

```
colors=[25, 35, 45, 55, 65]
```

```
sizes=[11, 22, 33, 44, 55]
```

```
plt.scatter(x, y, c=colors, s=sizes, cmap='viridis')
```

```
plt.colorbar()
```

```
# Histogram
```

```
marks=[90, 60, 50, 40, 30, 20, 80]
```

```
grade intervals=[0, 35, 70, 100]
```

```
plt.title('Student grade')
```

```
plt.hist(marks, grade intervals, histtype='stepfilled',  
         xwidth=0.7, facecolor='green')
```

```
plt.xticks([0, 35, 70, 100])
```

```
# bar
```

```
a=['a', 'b', 'c', 'd']
```

```
b=[20, 30, 40, 50]
```

```
# c=["red", "blue", "green", "black"]
```

```
plt.bar(a, b, width=0.3, color="green", ec="black")
```

```
# plt.bar(a, b, width=0.3, color=c, ec="black")
```

```
plt.xlabel('a values')
```

```
plt.ylabel('b values')
```

```
for index, value in enumerate(b):
```

```
    plt.text(index, value, str(value))
```



\_ / \_ / \_

```

x = ['2019', '2020', '2021', '2022', '2023']
y = [200, 300, 500, 150, 900]
c = ['blue', 'orange', 'yellow', 'red', 'green']
plt.bar(x, y, color=c, width=0.7)
plt.xlabel("Year")
plt.ylabel("Placements")
for index, value in enumerate(y):
    plt.text(index, value, sex(value))
# Stacked bar

```

```

x = ['CSE', 'AIML', 'DS', 'CS']
boys = [50, 40, 60, 70]
girls = [50, 60, 40, 30]
others = [10, 20, 30, 40]
plt.bar(x, boys, width=0.4, color='red', label='boys')
plt.bar(x, girls, width=0.4, bottom=boys, color='green',
        label='girls')
plt.bar(x, others, width=0.4, bottom=np.add(boys, girls),
        color='yellow', label='others')
plt.xlabel("Course")
plt.ylabel("Students")
plt.title("Student vs course")
plt.legend()

```



# pie chart

Student Performance = ['excellent', 'good', 'average', 'poor']

Student Values = [30, 60, 80, 100]

plt.figure(figsize=(8,10))

plt.pie(Student Values, labels=Student Performance,  
Startangle=90, explode=[0.2, 0.2, 0, 0],  
Shadow=True, colors=['red', 'green', 'blue', 'yellow'],  
autopct='%2.1f%%')

plt.legend(title='Performances')

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv('child Labour in India.csv')

df['Category'] = df['Category of States'].

replace(['Non special Category states',  
'Special Category States',  
['Non special', 'Special']])

df['Category']

df['Category'] == 'Special'

df[df['Category'] == 'Special']

Special = len(df[df['Category'] == 'Special'])

Special

non special = len(df[df['Category'] == 'Non special'])

non special



# Percentage of special, non special and All india under Category of States.

```
special = len(df[df['category'] == 'Special'])
non_special = len(df[df['category'] == 'Non special'])
All_india = len(df[df['category'] == 'All India'])
Sizes = [special, non_special, All_india]
labels = 'special', 'non special', 'All india'
colors = ['skyblue', 'yellow', 'green']
explode = (0.2, 0, 0.3)
plt.pie(Sizes, labels=labels, colors=colors,
        explode=explode, autopct='%1.1f%%', Shadow=True)
plt.title('Percentage of special, non special,
All india under category of States')
```

```
df['Manufacturing'] = df['Manufacturing'].replace('9.9', '9-9')
df['Manufacturing'] = df['Manufacturing'].astype('float')
```

# Stacked bar chart of child labour employee across various States.

```
fig, ax = plt.subplots(figsize=(17, 8))
ax.bar(df['States'], df['Agriculture'], color='red',
        label='Agriculture')
ax.bar(df['States'], df['Manufacturing'],
        bottom=df['Agriculture'], color='green',
        label='Manufacturing')
```



\_ / \_ / \_

```
ax.bar(df['States'], df['Construction'],  
       bottom=df['Agriculture'] + df['Manufacturing'],  
       color='blue', label='Construction')
```

```
ax.bar(df['States'], df['Trade Hotels & Restaurants'],  
       bottom=df['Agriculture'] + df['Manufacturing'] +  
       df['Construction'], color='7777',  
       label='Trade Hotels & Restaurants')
```

```
ax.bar(df['States'], df['Others'],  
       bottom=df['Agriculture'] + df['Manufacturing'] +  
       df['Construction'] + df['Trade Hotels & Restaurants'],  
       color='pink', label='Others')
```

```
plt.legend(bbox_to_anchor=(0, 1, 1, -0.102), loc=1,  
          ncol=3, mode="expand", borderaxespad=0.5)  
plt plt.xticks(rotation=90)  
plt.ylim(0, 110)
```