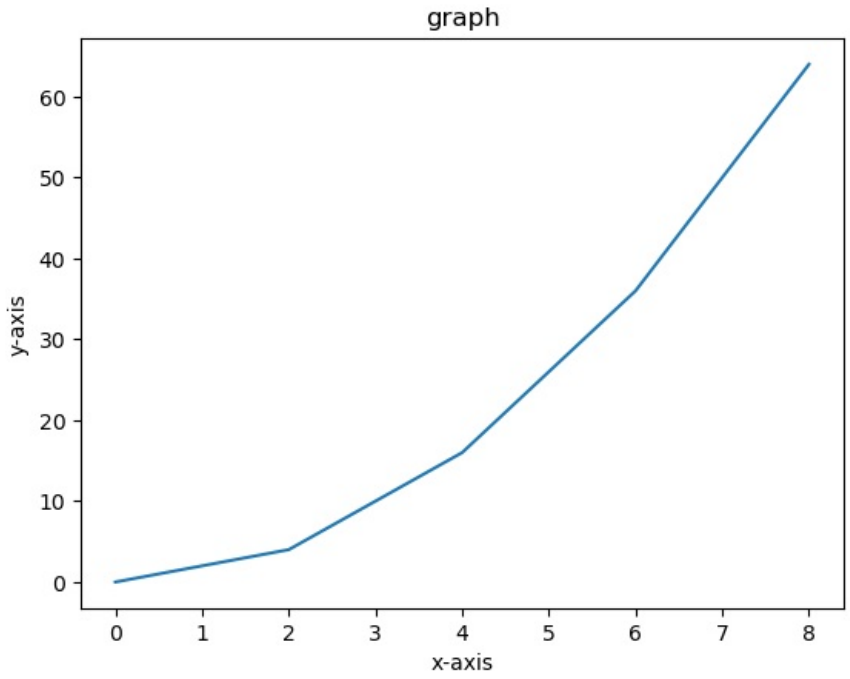
MATPLOTLIB

Matplotlib is a Python library for creating visualizations such as line plots, scatter plots, bar plots, histograms, and more. It provides a flexible and customizable interface to help you effectively visualize your data.

In [1]:
```
# plotting the simple line in matplotlib

import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**2

#making a line by plotting x and y
plt.plot(x,y)

#for setting title of graph
plt.title('graph')

#for x-label
plt.xlabel('x-axis')

#for y-label
plt.ylabel('y-axis')

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



A) MARKER (style, color, size)

- Representation of point in Graph/Chart is Marker

  Types of Marker styles

  - **' o ' = circle**
  - **' s ' = square**
  - **' d ' = diamond**
  - **' p ' = pentagon**

- ' h ' = hexagon
- ' . ' = point
- ' * ' = star
- ' + ' = plus
- ' | '= vertical line
- ' _ ' = horizontal line
- ' ^ ' = triangle_up marker
- ' < ' = triangle_left marker
- ' > '= triangle_right marker

if we use capital letter it becomes more bold like (Markers = 'D' instead of Marker='d')

for the Colour of the Markers

- we can use single letter predefined colours or use HEX ('#008000'), RGB or RGBA

The available pre-defined colours are:-

r = red

b = blue

w = white

k = black

c = cyan

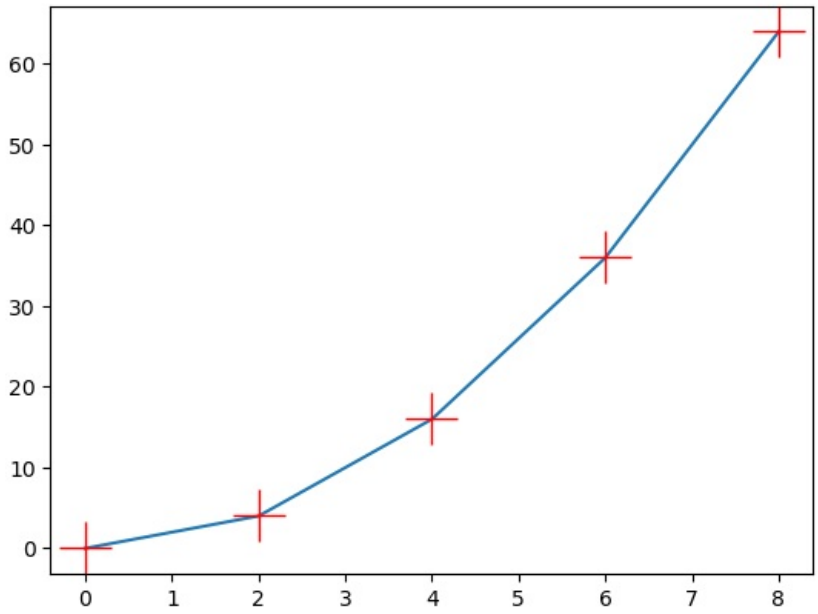
m = magenta

y = yellow

To use the marker and its colour and size we pass in plot() function as paramaters

- plt . plot (x , y , marker="value", mec="value", mfc="value", ms=value)
  - mec= marker edge colour
  - mfc= marker face colour
  - ms= marker size

In [2]:
```
import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**2

plt.plot(x,y,marker="+",mec="r",mfc="b",ms=25)
plt.show()

# mec= marker edge colour
# mfc= marker face colour
# ms= marker size
```



B) Lines (style, color, width)

1) line styles (ls) :-

- **solid**
- **dotted**
- **dashed**
- **dashdot**
- **none**

2) line width (lw) :-

- **lw="value"**

3) line color (c) :-

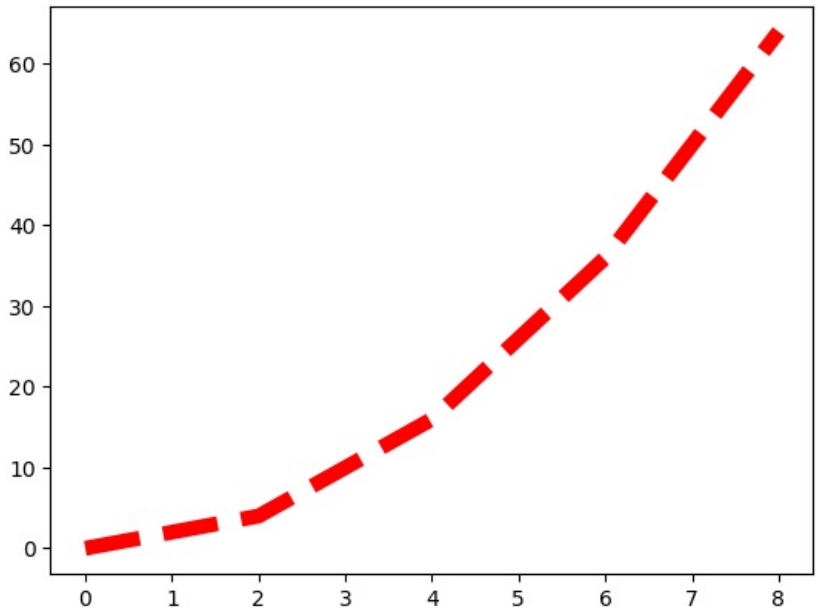
- **same colour as marker colors**

syntax:

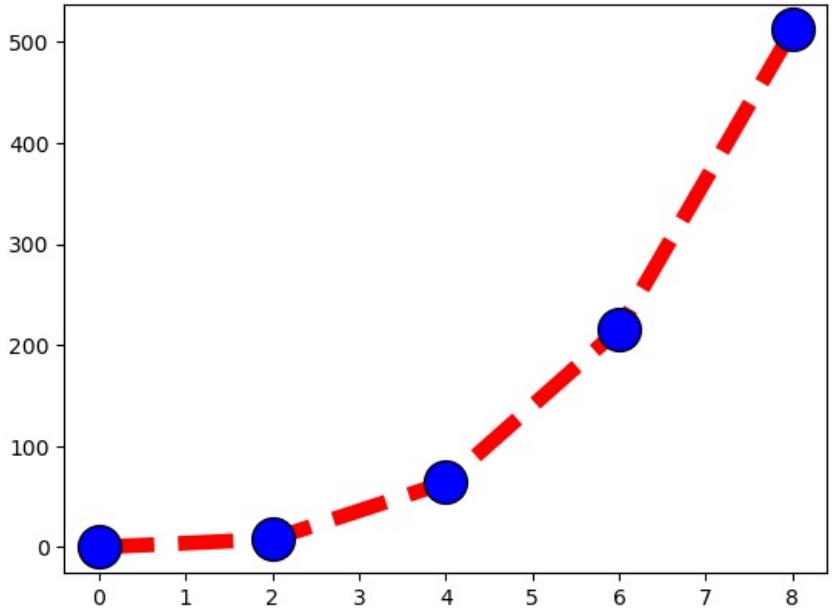
plt.plot (x, y) # gives default solid line

- **plt.plot (x, y, ls="value", lw=value, c="value")**

In [3]:
```
import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**2

plt.plot(x,y,ls="dashed",lw=7,c='r')
plt.show()
```



In [4]:
```
# using the markers and line properties at a same time
import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**3

plt.plot(x,y,marker='o',mec='k',mfc='b',ms=20,ls="dashed",lw=7,c='r')
plt.show()
```



C) Font properties for titles and lables (font, color, size, position(location), padding)

if we want to set the font ,color ,width and size to title and label we give value in fontdic as dictionary

- **fontdict={'font':Arial,'color':'r', 'size':20}**

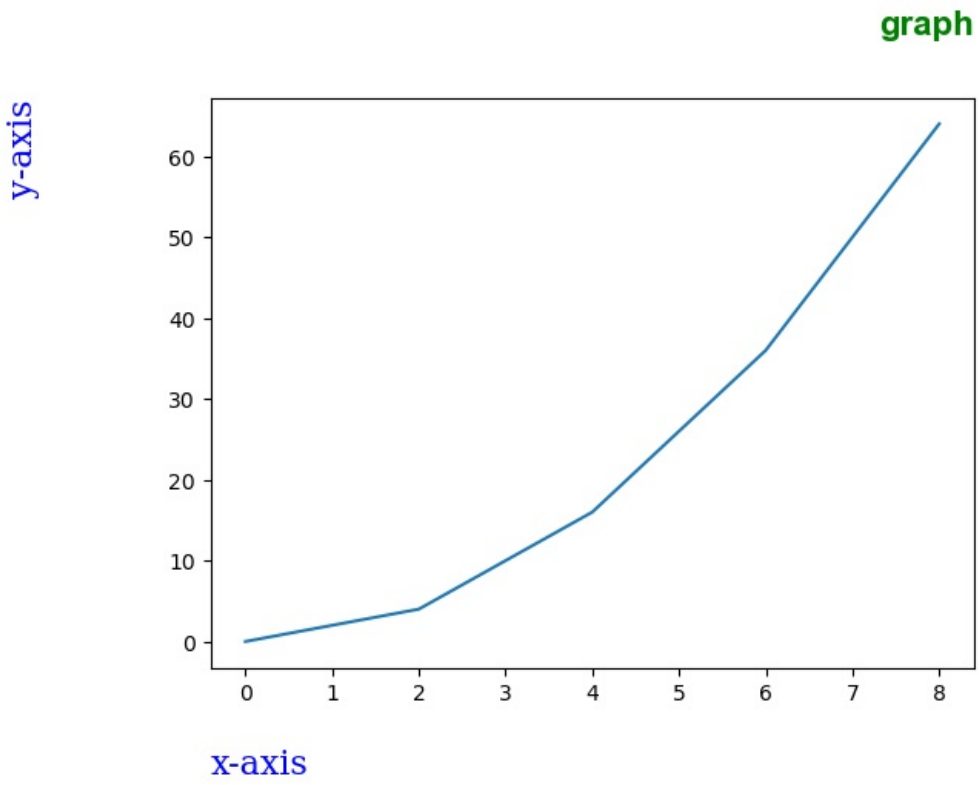
if we want to set postition(location) we give value to loc

- **for title and xlabel (we can give left, center and right as values in loc)**
- **for ylabel (we can give top, center and bottom as values in loc)**

if we want to give padding in title we can give values to (pad) and for x and y labels we can give values in (labelpad)

In [5]:
```
import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**2
```

```
#making a line by plotting x and y
plt.plot(x,y)

#to give value for fontdict
d={'font':'Arial','color':'g','size':16,'weight':'bold'}
d1={'font':'serif','color':'b','size':16}

#for setting title of graph
plt.title('graph',fontdict=d,loc='right',pad=30)

#for x-label
plt.xlabel('x-axis',fontdict=d1,loc='left',labelpad=20)

#plt.xlabel('x-axis',fontsize=15)

#for y-label
plt.ylabel('y-axis',fontdict=d1,loc='top',labelpad=60)

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

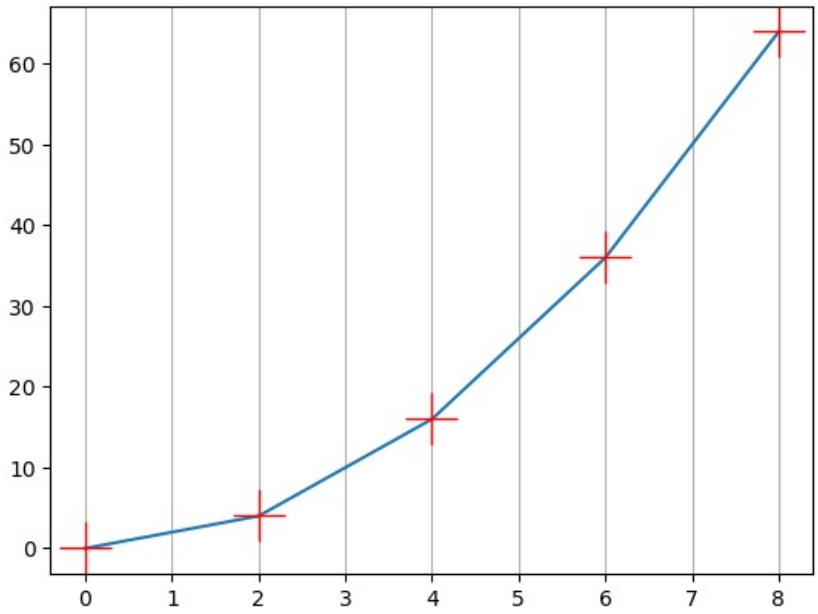


D) Grid Function grid()

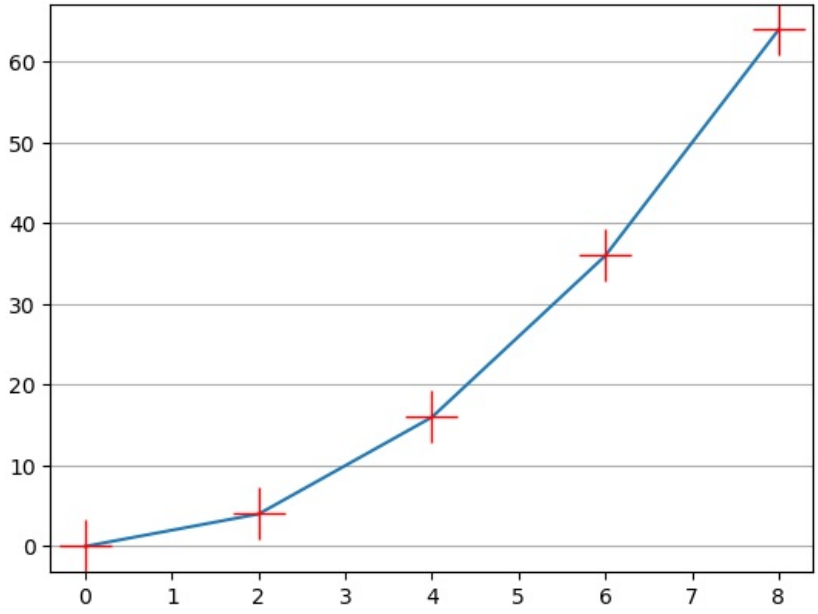
It generates the grid lines in the graph

- **for vertical lines through x axis --- axis='x'**
- **for horizontal lines through y axis --- axis='y'**
- **for both --- axis='both'**

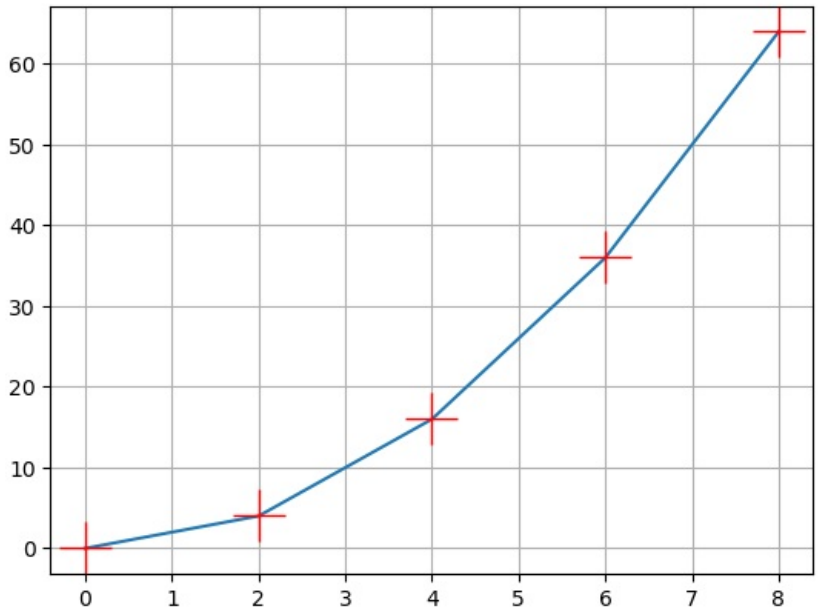
In [6]:
```
#for vertical lines through x axis

import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**2

plt.plot(x,y,marker="+",mec="r",mfc="b",ms=25)
plt.grid(axis='x')
plt.show()
```



In [7]:
```
#for horizontal lines through y axis

import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**2

plt.plot(x,y,marker="+",mec="r",mfc="b",ms=25)
plt.grid(axis='y')
plt.show()
```



In [8]:
```
# for both axis

import matplotlib.pyplot as plt # or from matplotlib import pyplot as plt
import numpy as np

x=np.arange(0,10,2)
y=x**2

plt.plot(x,y,marker="+",mec="r",mfc="b",ms=25)
plt.grid(axis='both')
plt.show()
```



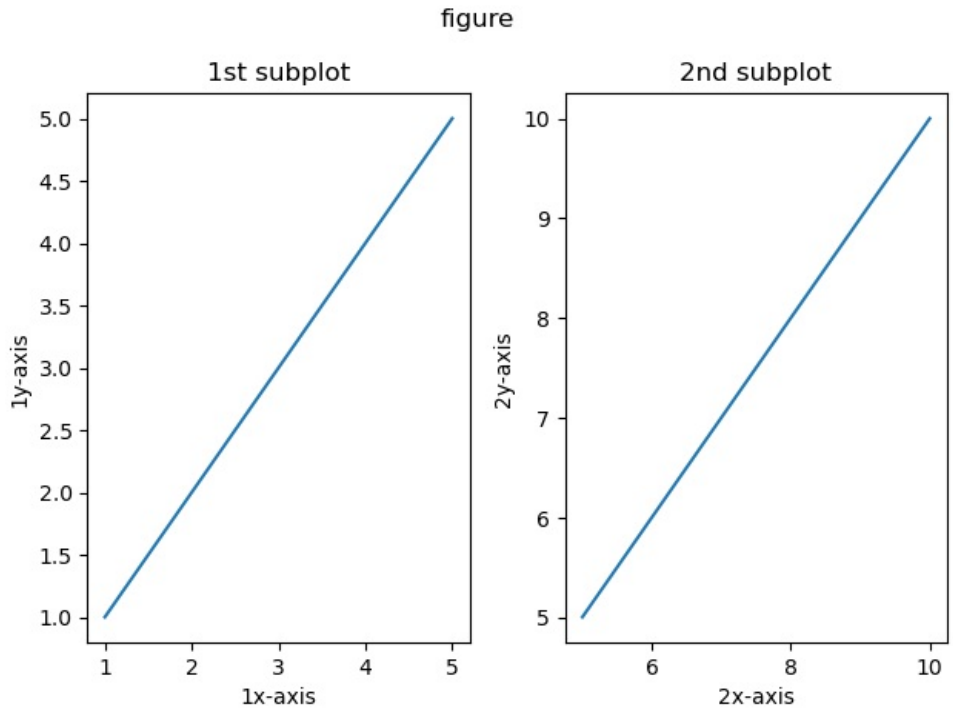
E) SUBPLOTS

- **we can make many subplot in a single figure to make subplots we can we can use subplot(row, column, index)**

  suppose we want two subplots in a single row we can use row= 1 column =2
  suppose we want three subplots in a single column we can use row= 3 column =1

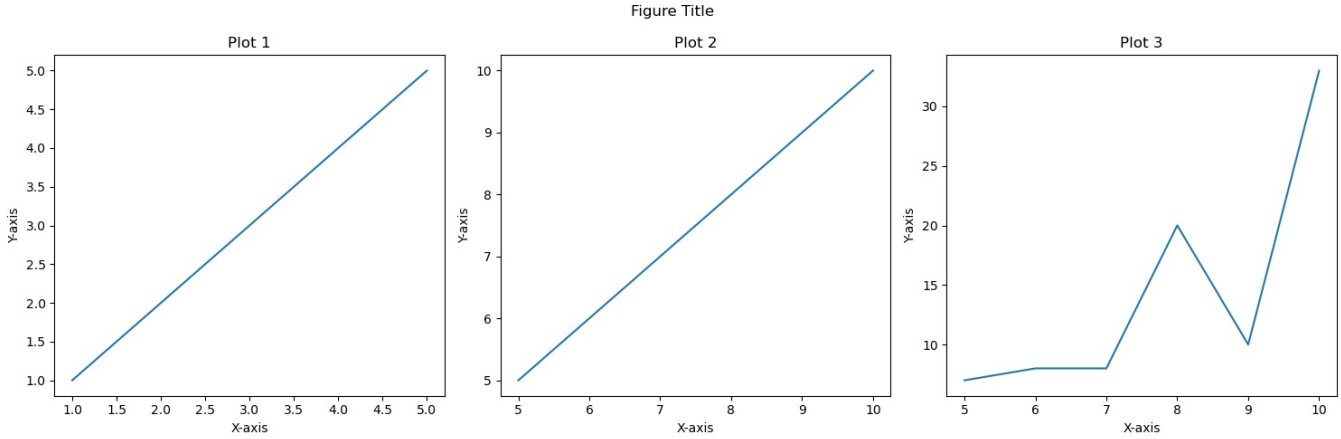
  - **the number of plots is given by multiplying the value of row * column**
  - **we use suptitle() to give title to main heading**
  - **plt.tight_layout() is used to avoid overlapping between figures**

In [9]:
```
#making two subplots 1st method

from matplotlib import pyplot as plt
x1=[1,2,3,4,5]
y1=[1,2,3,4,5]

x2=[5,6,7,8,9,10]
y2=[5,6,7,8,9,10]

plt.subplot(1,2,1)# euta row dui ota column ko plot xa tesma 1st postition ko subplot yo ho vaneko
plt.plot(x1,y1)
plt.title("1st subplot")
plt.xlabel("1x-axis")
plt.ylabel("1y-axis")


plt.subplot(1,2,2)# euta row dui ota column ko plot xa tesma 2nd postition ko subplot yo ho vaneko
plt.plot(x2,y2)
plt.title("2nd subplot")
plt.xlabel("2x-axis")
plt.ylabel("2y-axis")

plt.suptitle("figure")
plt.tight_layout() # to avoid the overlapping we can use
plt.show()
```

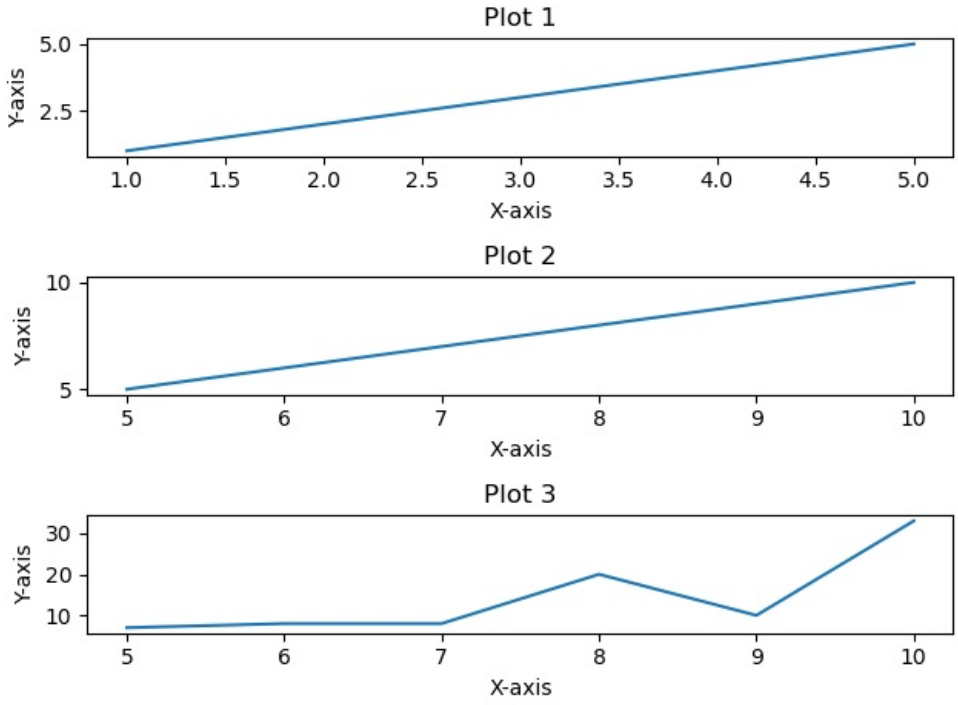


In [10]:

```
import matplotlib.pyplot as plt

x1 = [1, 2, 3, 4, 5]
y1 = [1, 2, 3, 4, 5]

x2 = [5, 6, 7, 8, 9, 10]
y2 = [5, 6, 7, 8, 9, 10]

x3 = [5, 6, 7, 8, 9, 10]
y3 = [7, 8, 8, 20, 10, 33]

# Create subplots
fig, axs = plt.subplots(1, 3, figsize=(15, 5))  # Adjust figsize as needed

# Flatten the array of axes objects
axs = axs.flatten()

# Plot on each subplot
axs[0].plot(x1, y1)
axs[0].set_title("Plot 1")
axs[0].set_xlabel("X-axis")
axs[0].set_ylabel("Y-axis")

axs[1].plot(x2, y2)
axs[1].set_title("Plot 2")
axs[1].set_xlabel("X-axis")
axs[1].set_ylabel("Y-axis")

axs[2].plot(x3, y3)
axs[2].set_title("Plot 3")
axs[2].set_xlabel("X-axis")
axs[2].set_ylabel("Y-axis")

# Add overall title and adjust layout
plt.suptitle("Figure Title")
plt.tight_layout()

# Display the plots
plt.show()
```

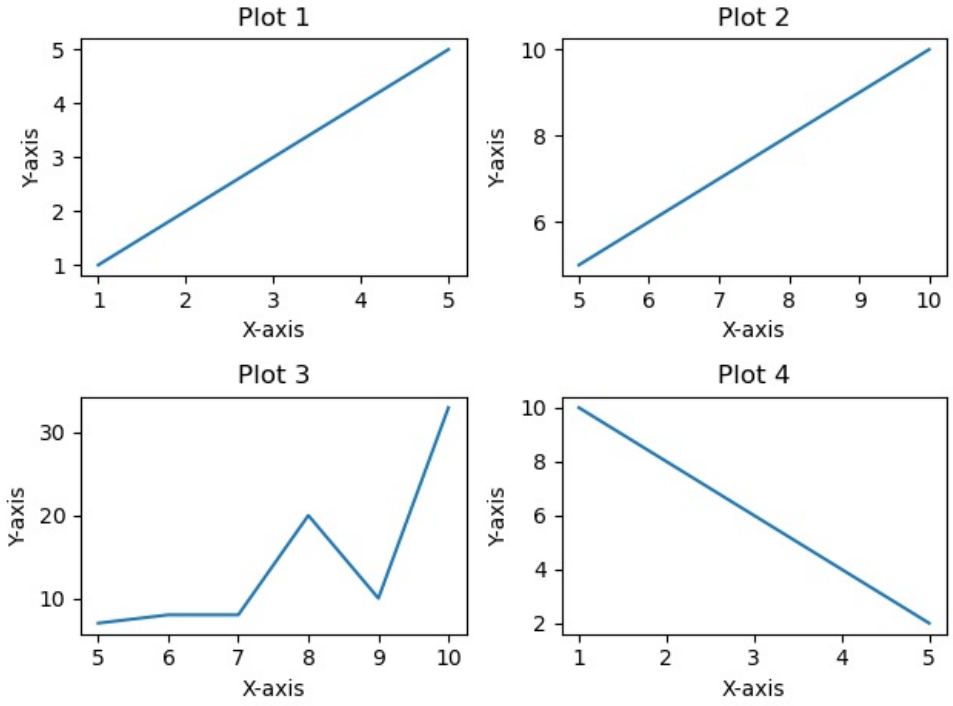


In [11]:
```
#if we have value 1 for row or column plt.subplots(1, 3), returns a NumPy array of 1D SO we have to access ax[0
#ax[1].plot(x2, y2),ax[2].plot(x3, y3) like this


'''here column=1 plt.subplots(1, 3)'''
from matplotlib import pyplot as plt

x1 = [1, 2, 3, 4, 5]
y1 = [1, 2, 3, 4, 5]

x2 = [5, 6, 7, 8, 9, 10]
y2 = [5, 6, 7, 8, 9, 10]

x3 = [5, 6, 7, 8, 9, 10]
y3 = [7, 8, 8, 20, 10, 33]

fig, ax = plt.subplots(3, 1)

ax[0].plot(x1, y1)
ax[0].set_title("Plot 1")
ax[0].set_xlabel("X-axis")
ax[0].set_ylabel("Y-axis")

ax[1].plot(x2, y2)
ax[1].set_title("Plot 2")
ax[1].set_xlabel("X-axis")
ax[1].set_ylabel("Y-axis")

ax[2].plot(x3, y3)
ax[2].set_title("Plot 3")
ax[2].set_xlabel("X-axis")
ax[2].set_ylabel("Y-axis")

plt.tight_layout()

plt.show()
```



In [12]:
```
from matplotlib import pyplot as plt

# Data for the subplots
x1 = [1, 2, 3, 4, 5]
y1 = [1, 2, 3, 4, 5]

x2 = [5, 6, 7, 8, 9, 10]
y2 = [5, 6, 7, 8, 9, 10]

x3 = [5, 6, 7, 8, 9, 10]
y3 = [7, 8, 8, 20, 10, 33]

x4 = [1, 2, 3, 4, 5]
y4 = [10, 8, 6, 4, 2]

# Create the subplots
fig, ax = plt.subplots(2, 2)

# Flatten the array of axes objects
ax = ax.flatten()
```

```
# Plot on each subplot
ax[0].plot(x1, y1)
ax[0].set_title("Plot 1")
ax[0].set_xlabel("X-axis")
ax[0].set_ylabel("Y-axis")

ax[1].plot(x2, y2)
ax[1].set_title("Plot 2")
ax[1].set_xlabel("X-axis")
ax[1].set_ylabel("Y-axis")

ax[2].plot(x3, y3)
ax[2].set_title("Plot 3")
ax[2].set_xlabel("X-axis")
ax[2].set_ylabel("Y-axis")

ax[3].plot(x4, y4)
ax[3].set_title("Plot 4")
ax[3].set_xlabel("X-axis")
ax[3].set_ylabel("Y-axis")

plt.tight_layout()
plt.show()
```



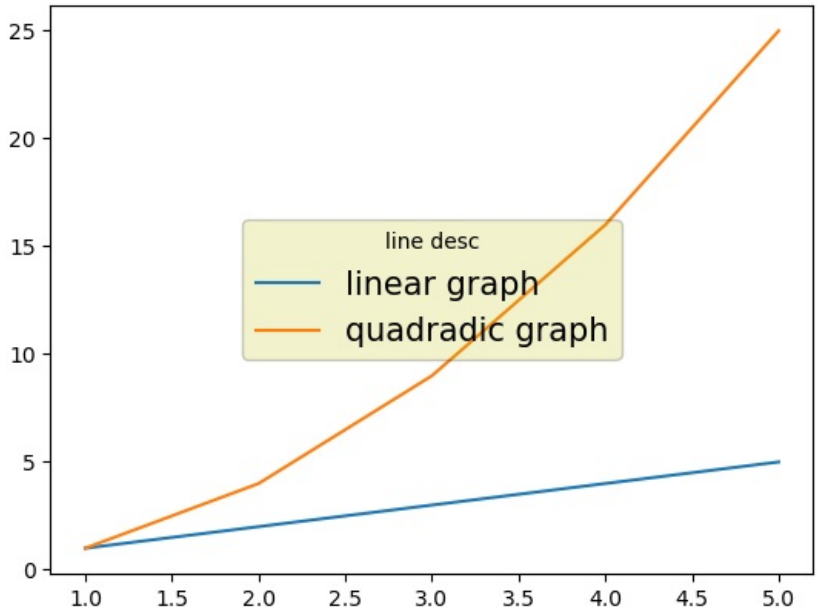
E) Legend

legend function parameters:-

- **label= we pass list in label inside legend function for how many curve description we want according to our curves**
- **loc= we can give 'best', 'upper right', 'upper left', 'lower left', 'lower right', 'right', 'center left', 'center right', 'lower center', 'upper center', 'center' for position of legend box**
- **fontsize=value**
- **facecolor='g' (gives the background color to box)**
- **edgecolor='b' (give the color to the edge)**
- **framealpha=value (value between 0-1) for opacity of the frame**

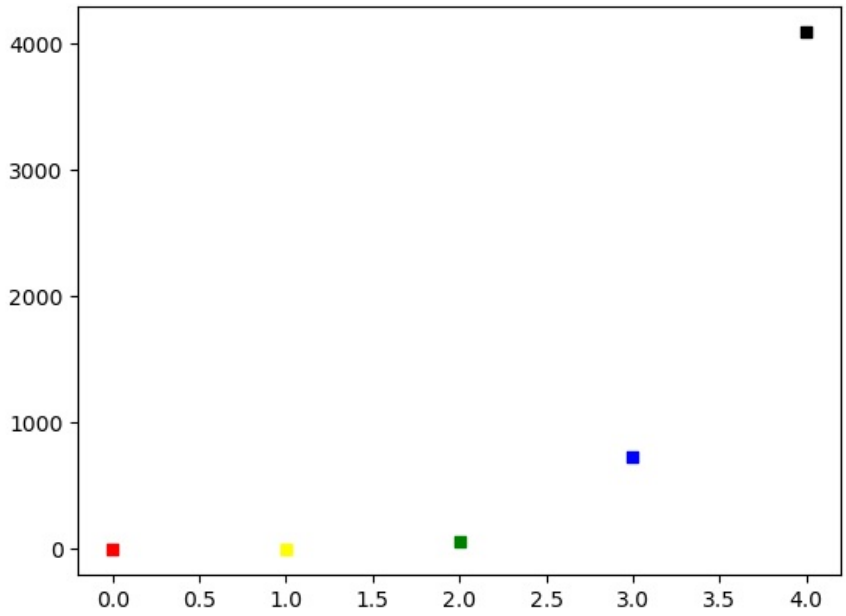
In [13]:
```
import matplotlib.pyplot as plt

x1 = [1, 2, 3, 4, 5]
y1 = [1, 2, 3, 4, 5]

x2=[1, 2, 3, 4, 5]
y2=[1,4,9,16,25]
```

```
plt.plot(x1,y1,x2,y2)
plt.legend(title="line desc",labels=['linear graph','quadradic graph'],loc="center",fontsize=15,
           facecolor='y',edgecolor='k',framealpha=0.2)
plt.show()
```



F) SCATTER PLOT

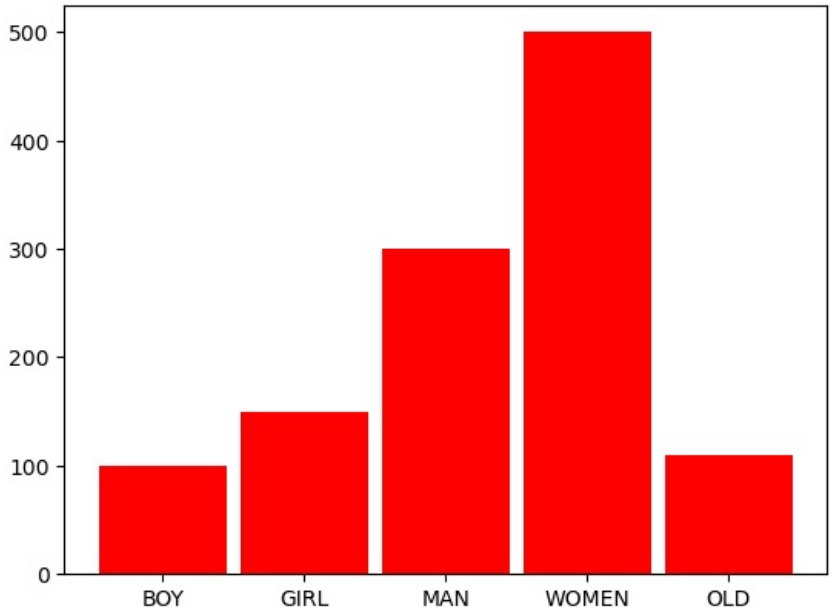
In [14]:
```
# making scatterplot in matplotlib
import matplotlib.pyplot as plt
import numpy as np
x=np.arange(0,5)
y=x**6

#plt.scatter(x,y,marker='s',c="m",s=25)

'''we can also pass the color in list for c'''

col=['red','yellow','green','blue','black']
plt.scatter(x,y,marker='s',c=col,s=25)

# parameters info
# c for color, s for size of marker in scatter plot

plt.show()
```



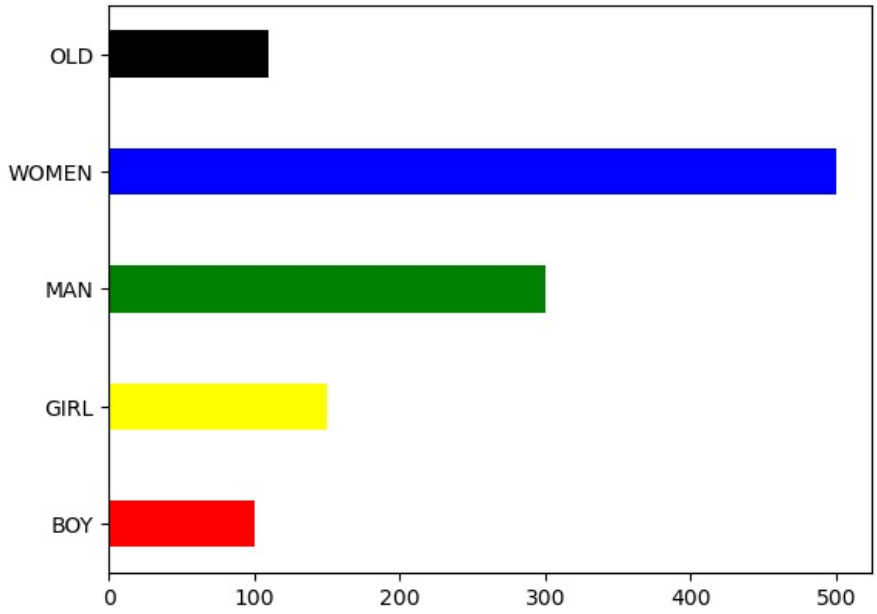
G) BAR GRAPH (bar(), barh())

In [15]:
```
# plotting the vertical bar graph with single color

from matplotlib import pyplot as plt
```

```
x=['BOY','GIRL','MAN','WOMEN','OLD']
y=[100,150,300,500,110]

#for plotting bar graph vertical we use bar()

plt.bar(x,y,color='r',width=0.9) #for horizontal bar we use height instead of width
plt.show()
```



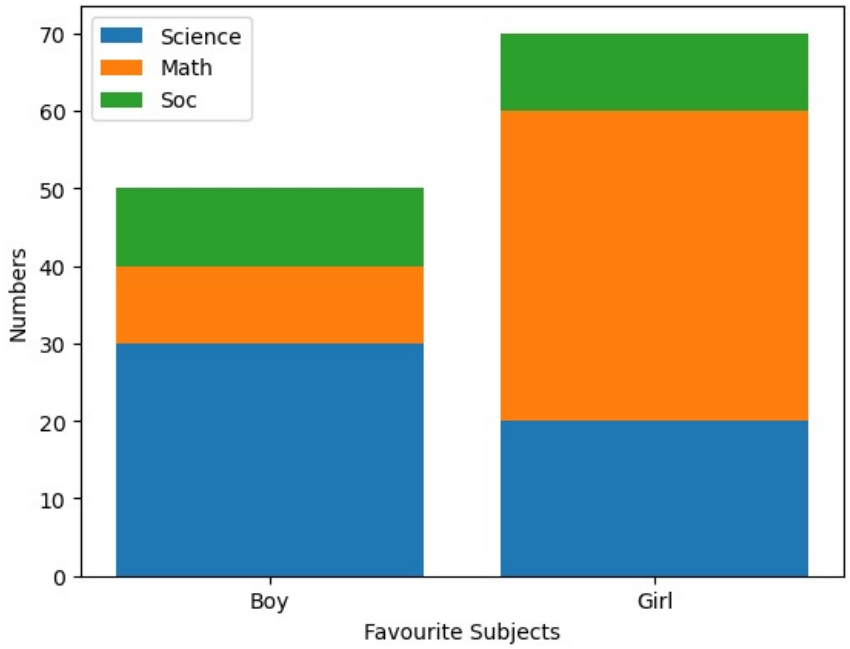
In [16]:
```
# plotting the horizontal bar graph with multiple color

from matplotlib import pyplot as plt

x=['BOY','GIRL','MAN','WOMEN','OLD']
y=[100,150,300,500,110]

#for plotting bar graph vertical we use bar()
col=['red','yellow','green','blue','black']

plt.barh(x,y,color=col,height=0.4)  # passing list in c and using height for thickness of horizontal bar diagra

plt.show()
```



H) PLOTTING THE STACKED BAR DIAGRAM

In [17]:
```
from matplotlib import pyplot as plt
import numpy as np

#favoutite subject
x=['Boy','Girl']
sci=[30,20]
math=[10,40]
soc=[10,10]
```

```
plt.bar(x,sci)# yo bottom bata suru hune vayo yesko end paxi aeko suru garna prxa so,
plt.bar(x,math,bottom=sci)# sci sake paxi math suru vho
plt.bar(x,soc,bottom=np.add(sci,math)) # sci ra math sake paxi yo suru hunu prxa so bottom= sci+math hunu paryo
plt.xlabel("Favourite Subjects")
plt.ylabel("Numbers")
plt.legend(labels=["Science","Math","Soc"])
plt.show()

"""
Boy ko ma herem---->

suru ma science ko 0 bata dekhayo
aba math ko lagi science  sake paxi dekhaunu paryo 30 bata suru garna yo code chaiyoo plt.bar(x,math,bottom=sci
aba social ko lagi sci ani math sakeko tham paxi suru hunu paryo so sci+math
sci=[30,20]
math=[10,40]
soc=[10,10]

soc ko lagi suru point= 30+10 hunu paryo girl ko lagi ani girl ko lagi 40+20
--->so naya array numpy.add ley yo dinxa [40,60]

"""
# look in graph
#boy ---> sci=30 , math = (40-30) = 10 , soc = (50-40) = 10
#girl ---> sci=20 , math = (60-20) = 40 , soc = (70-60) = 10
```

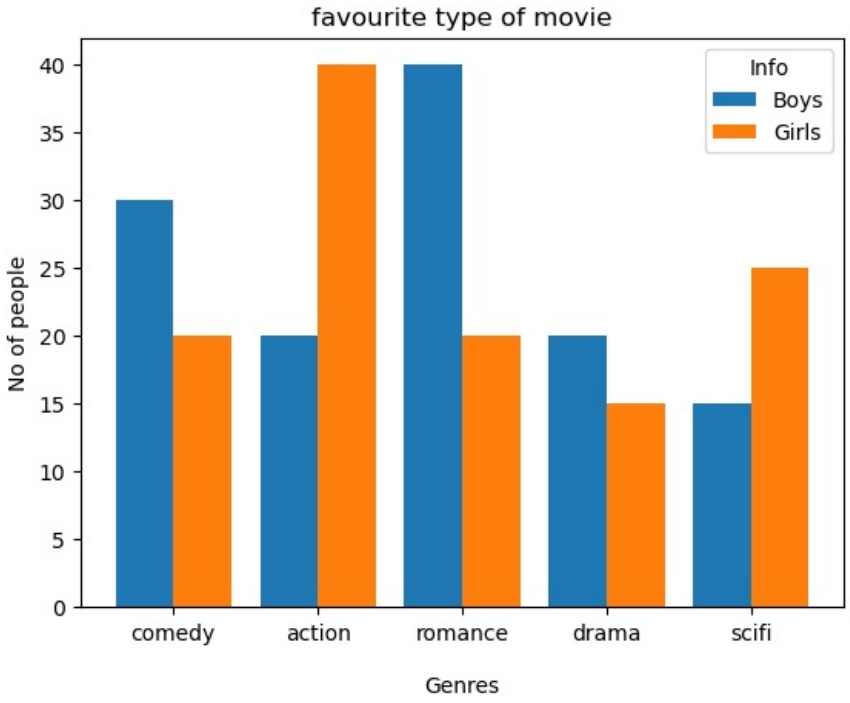


Out[17]:
```
'\nBoy ko ma herem---->\n\nsuru ma science ko 0 bata dekhayo \naba math ko lagi science  sake paxi dekhaunu par
yo 30 bata suru garna yo code chaiyoo plt.bar(x,math,bottom=sci) \naba social ko lagi sci ani math sakeko tham
paxi suru hunu paryo so sci+math\nsci=[30,20]\nmath=[10,40]\nsoc=[10,10]\n\nsoc ko lagi suru point= 30+10 hunu
paryo girl ko lagi ani girl ko lagi 40+20\n--->so naya array numpy.add ley yo dinxa [40,60]\n\n'
```

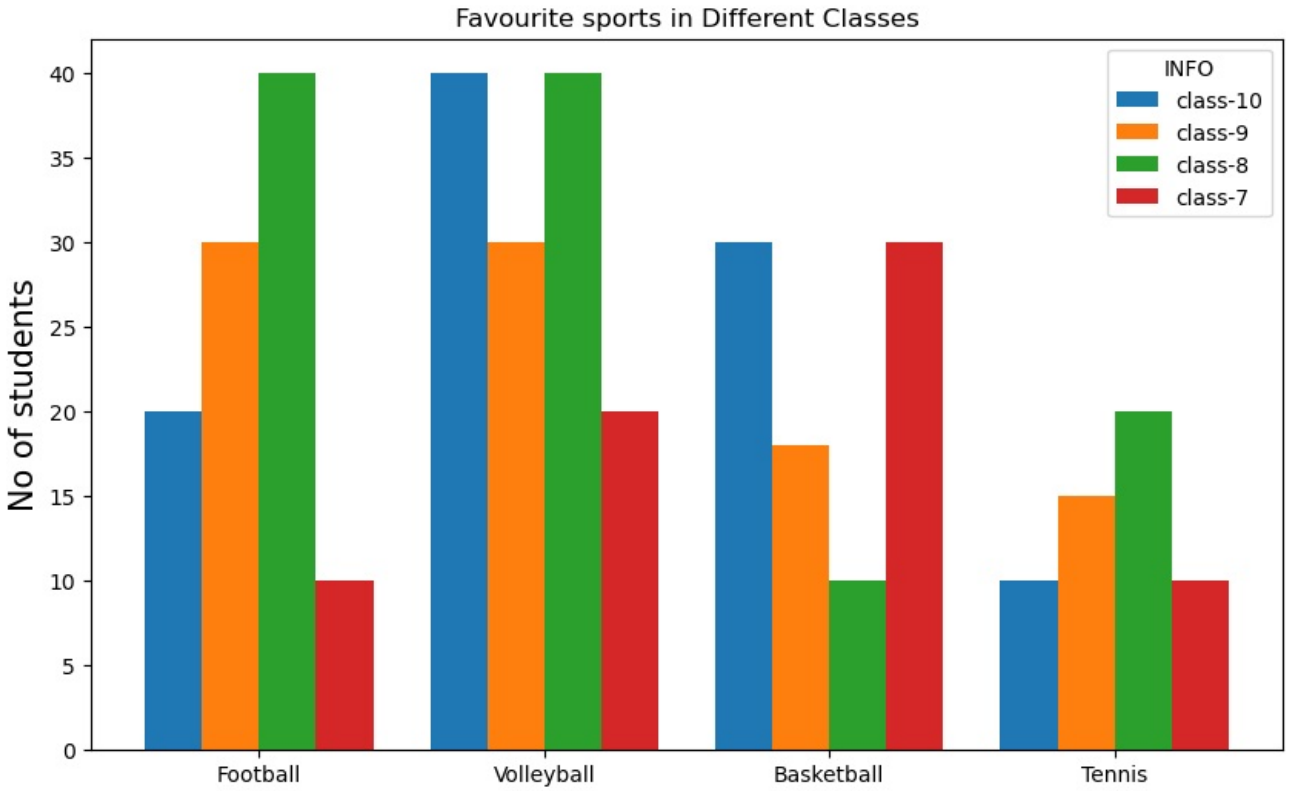
I) PLOTTING MULTIPLE BAR DIAGRAM

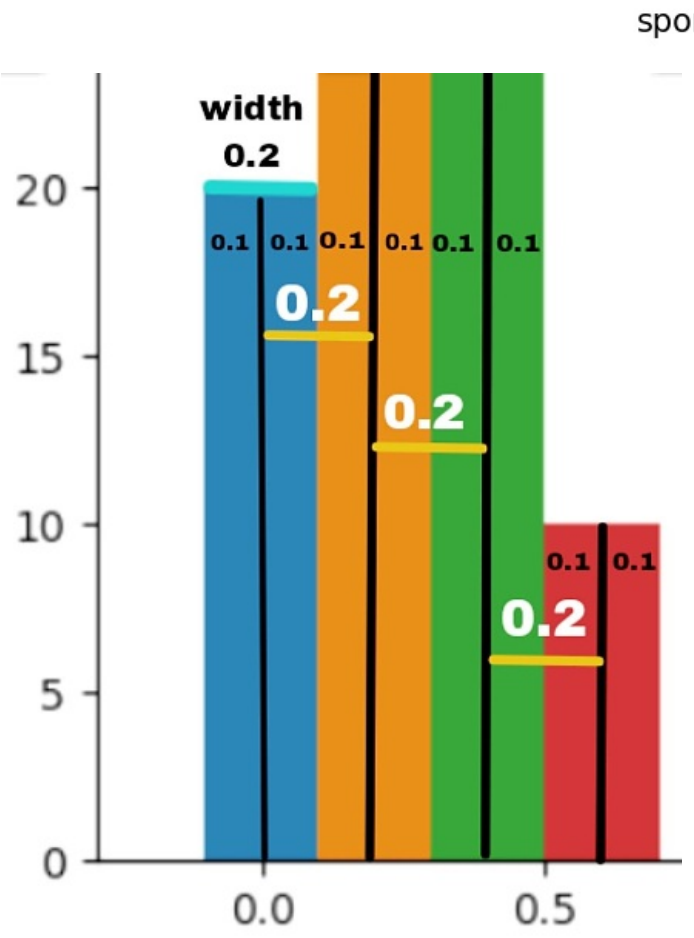
In [18]:
```
import matplotlib.pyplot as plt
import numpy as np

genres=['comedy','action','romance','drama','scifi']
boys=[30,20,40,20,15]
girls=[20,40,20,15,25]

values=np.arange(len(genres)) #give list from 0 - 4

#width of the bar
width=0.4

plt.bar(values,boys,width)

plt.bar(values+width,girls,width)

plt.xlabel("Genres",labelpad=15)

plt.ylabel("No of people")

plt.title("favourite type of movie")

plt.legend(title="Info",labels=['Boys','Girls'])

plt.xticks(values+(width/2),genres)
```

```
plt.show()
```



In [19]:
```
import matplotlib.pyplot as plt
import numpy as np


sports=['Football','Volleyball','Basketball','Tennis']

class_10=[20,40,30,10]
class_9=[30,30,18,15]
class_8=[40,40,10,20]
class_7=[10,20,30,10]

width=0.2

num_values=np.arange(len(sports))

plt.figure(figsize=(10, 6))# Width: 10 inches, Height: 6 inches

plt.bar(num_values,class_10,width)
plt.bar(num_values+width, class_9,width)
plt.bar(num_values+width+width,class_8,width)
plt.bar(num_values+3*(width),class_7,width)
plt.xlabel("sports",labelpad=25,fontsize=15)
plt.ylabel("No of students",fontsize=15)

plt.xticks(num_values+((3*width)/2),sports) #num_values+ width lai sports label ley replace gareko
# ani label bich ma launa ko lagi  numvalues + (last kati ota width xa)/2 garnu prxa trick

plt.title("Favourite sports in Different Classes")
plt.legend(title="INFO",labels=['class-10','class-9','class-8','class-7'])


plt.show()
```





plt.bar(num_values,class_10,width)
plt.bar(num_values+width, class_9,width)
plt.bar(num_values+width+width,class_8,width)
plt.bar(num_values+3*(width),class_7,width)

num_values = [0,1,2,3]

lets see the football man parauni graph

---- first blue bar x-axis ko 0 ma aauni vayo

---- second orange bar 0+width(0+0.2) bata suru huni vayo

---- third green bar 0+width+width(0+0.2+0.2) baat suru huni vayo

---- fourth red bar 0+width+width+width or 3*(width) bata suru huni vayo

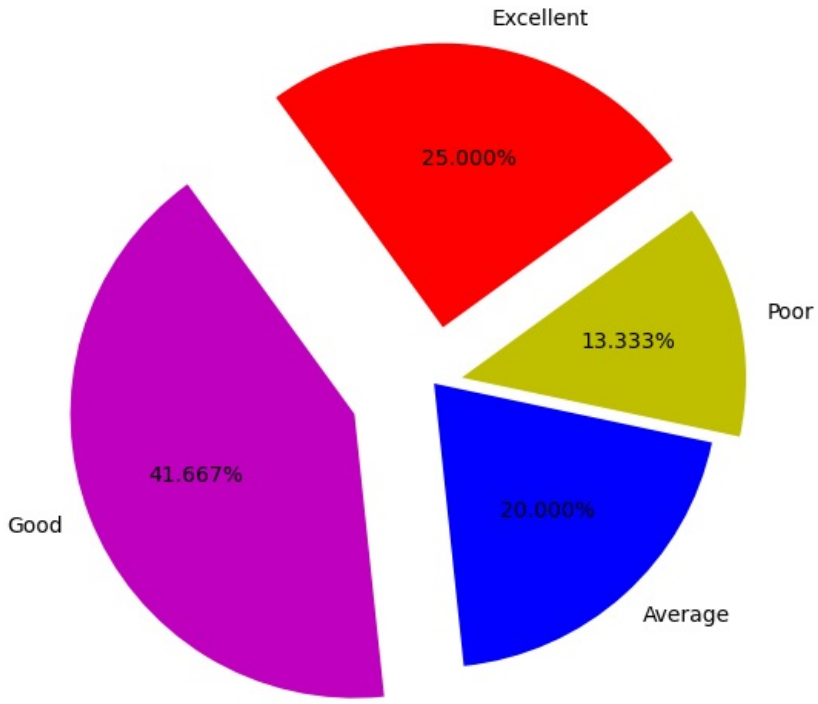
J) PLOTTING A PIE CHART

In [20]:
```
#plotting the piechart
from matplotlib import pyplot as plt

student_performance=["Excellent","Good","Average","Poor"]
student_values=[15,25,12,8]
plt.pie(student_values)
plt.show()
```

PARAMETERS

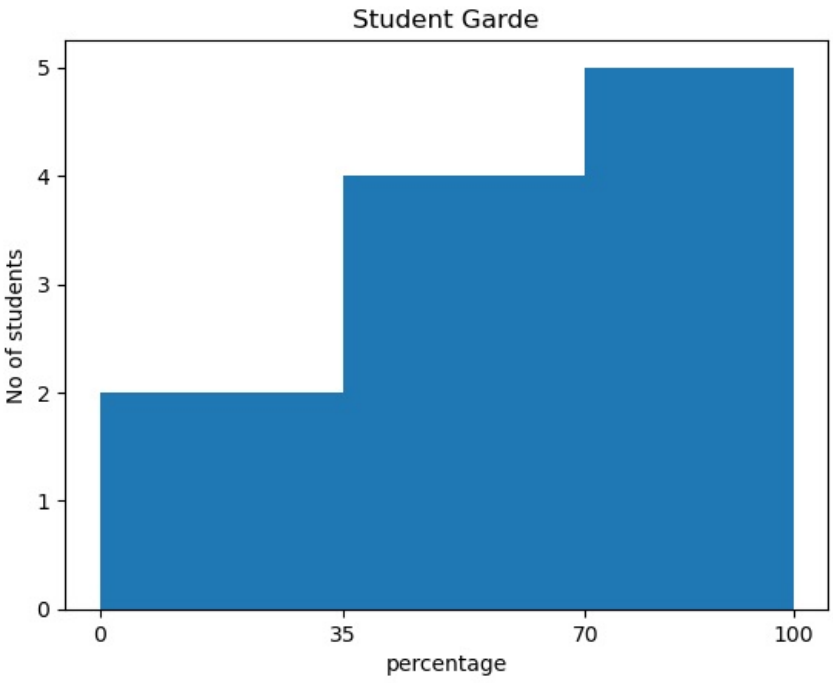
- labels=[] (for the information of portion of pie)
- startangle=0,90,180,270,360 any degree(at what angle from the pie start to begin) then moves counterclock wise
- explode=[value, value, value ,value] (takes out the portion from pie)
- colors=[value, value, value ,value] (gives colors to the pie)
- autopct=" %.3f%%" (it show percentage to pie portion decimal value paxi 3 ota num dekhaucxa) # no space between %%

In [21]:
```
#plotting the piechart
from matplotlib import pyplot as plt

student_performance=["Excellent","Good","Average","Poor"]
student_values=[15,25,12,8]
c=['r','m','b','y']
plt.figure(figsize=(10, 6))# Width: 10 inches, Height: 6 inches
plt.pie(student_values,labels=student_performance,startangle=36,explode=[0.2,0.3,0,0.1],colors=c,autopct="%.3f%

plt.show()
```



K) PLOTTING HISTOGRAM

In [22]:
```
from matplotlib import pyplot as plt
marks=[90,80,80,63,83,23,43,22,55,43,100]
grade_intervals=[0,35,70,100] #(0-35,35-70,70-100) yo interval ma kati jana parxa number dinxa
plt.title("Student Garde")
plt.hist(marks,grade_intervals) #y axis  ma interval ma kti jana prxa number dixa
plt.xticks([0,35,70,100],grade_intervals) #changing the x ticks
plt.xlabel("percentage")
plt.ylabel("No of students")
plt.show()
```



PARAMETERS

- **histtype='bar','step','stepfilled'**

- width=value (0-1) only for bar histogram
- facecolor="value"
- edgecolor="value"

In [23]:
```
from matplotlib import pyplot as plt

marks=[90,80,80,63,83,23,43,22,55,43,100]

grade_intervals=[0,35,70,100] #(0-35,35-70,70-100) yo interval ma kati jana parxa number dinxa

plt.title("Student Garde")

plt.hist(marks, grade_intervals, histtype='bar', rwidth=0.95, facecolor='red'
        ,edgecolor='k')#y axis  ma interval ma kti jana prxa number dixa

plt.xticks([0,35,70,100]) #changing the x ticks

plt.xlabel("percentage")

plt.ylabel("No of students")

plt.show()
```