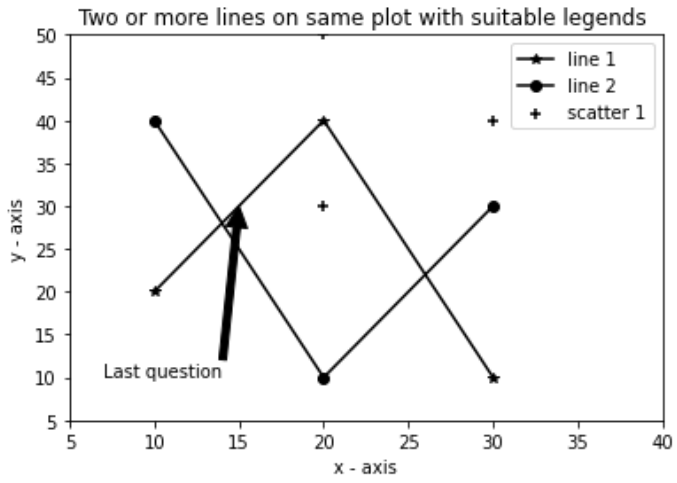
1. Please plot the figures described by the code below. The interval of x and y axis is 5. Correct graph type, Correct label, Correct x, y boundaries, Correct chart title, Show legend, Correct annotations. Correct marker.

```
import matplotlib.pyplot as plt
x1 = [10,20,30]
y1 = [20,40,10]
plt.plot(x1, y1, label = "line 1", marker = "*", color = "black")
x2 = [10,20,30]
y2 = [40,10,30]
plt.plot(x2, y2, label = "line 2", marker = "o", color = "black")
x3 = [20, 20, 30]
y3 = [30, 50, 40]
plt.scatter(x3, y3, label = "scatter 1", marker = "+", color = "black")
plt.xlabel('x - axis')
plt.ylabel('y - axis')
plt.xlim((5,40))
plt.ylim((5,50))
plt.title('Two or more lines on same plot with suitable legends ')
plt.annotate('Last question', xy=(15, 30), xytext=(7, 10), arrowprops=dict(facecolor='black
', shrink=0.0002))
plt.legend()
plt.show()
```



2. A matrix with checkboard patterns looks like below:

```
Checkerboard pattern:
[[0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]]
```

Please create a 80 × 80 NumPy matrix with the above checkboard pattern.

Solution:

```
import numpy as np
print("Checkerboard pattern:")
x = np.zeros((80,80),dtype=int)
x[1::2,::2] = 1
x[::2,1::2] = 1
print(x)
```

3. Please show the matrix defined by the code below.

```
a = int(np.arange(7).mean())
maxvalue = int(np.arange(7).max())
b = np.ones([a, a])
t = np.pad(b, (2, 2), 'constant', constant_values=0)
width = t.shape[1] ##This is to get the width of t (its second dimension)
for index in range(width):
    t[index, -index] = maxvalue
print(t)
```

4. Please write a program to create a matrix with the following forms:

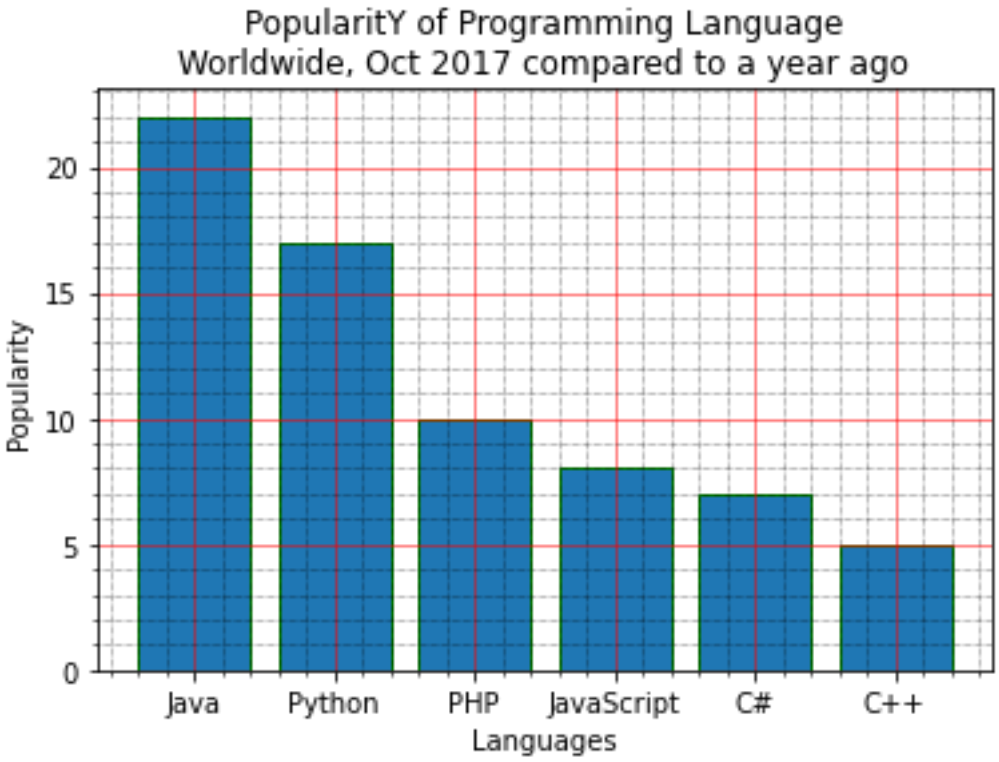
```
[[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]
 [ 1.  3.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. 15.]
 [ 2.  0.  6.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. 14.  1.]
 [ 3.  0.  0.  9.  0.  0.  0.  0.  0.  0.  0.  0. 13.  0.  1.]
 [ 4.  0.  0.  0. 12.  0.  0.  0.  0.  0.  0. 12.  0.  0.  1.]
 [ 5.  0.  0.  0.  0. 15.  0.  0.  0.  0. 11.  0.  0.  0.  1.]
 [ 6.  0.  0.  0.  0.  0. 18.  0.  0. 10.  0.  0.  0.  0.  1.]
 [ 7.  0.  0.  0.  0.  0.  0. 21.  9.  0.  0.  0.  0.  0.  1.]
 [ 8.  0.  0.  0.  0.  0.  0.  8. 24.  0.  0.  0.  0.  0.  1.]
 [ 9.  0.  0.  0.  0.  0.  7.  0.  0. 27.  0.  0.  0.  0.  1.]
 [10.  0.  0.  0.  0.  6.  0.  0.  0.  0. 30.  0.  0.  0.  1.]
 [11.  0.  0.  0.  5.  0.  0.  0.  0.  0.  0. 33.  0.  0.  1.]
 [12.  0.  0.  4.  0.  0.  0.  0.  0.  0.  0.  0. 36.  0.  1.]
 [13.  0.  3.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. 39.  1.]
 [14.  2.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]]
```

Requirements: It is not allowed to create the matrix by writing down each element.

```
import numpy as np
matrix = np.zeros([15, 15])
for x in range(15):
      matrix[x, x]=x*3
      matrix[-x,x]=x+1
      matrix[x, 0]=x
      matrix[x, 14] = 1

print(matrix)
```

5. Please complete the following code to display the figures with the following style:




```
import matplotlib.pyplot as plt
    #Write your code in this place. The grid code below has plotted the grid
    in the figure. Please finish other parts. You need to define the X, Y axis,  Correct label,
    Correct chart title, and use an appropriate to plot the this bar.


# Turn on the grid
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')
# Customize the minor grid
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```

Solutions:

```
import matplotlib.pyplot as plt
x = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22, 17, 10, 8, 7, 5]

plt.bar(x, popularity, edgecolor='green')

plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.title("PopularitY of Programming Language\n" + "Worldwide, Oct 2017 c
ompared to a year ago")

# Turn on the grid
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')
# Customize the minor grid
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```

6. We have discussed the example below in our class. Now, please answer the following questions based on its variants.
   Please plot ALL figures displayed in the console based on the code below:
   Requirement: Correct x y axis range limitation is required based on the description, correct figure style

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, clear_output

fig = plt.figure()
x = np.array([2, 5, 8, 10, 12, 23])
y = x**2

ax = fig.subplots()
l, = ax.plot(x, y, marker = "*")

ax.axis(xmin=0,xmax=30) # set the range of x axis from 0 to 30
ax.axis(ymin=0,ymax=100) # set the range of y axis from 0 to 100


def animate(i):
    l.set_data(x[:i], y[:i])
    return l

for i in range(len(x)):
    print(i)
    animate(i)
    display(fig)
    if i==3:
      break
```

7. Supposing there is a csv file named as “salary.csv”, which stores the salary information of the employers in departments A,B, and C. The first 7 rows are shown as follows to indicate the details of the data.

staff_id	department	name	gender	salary
1	A	sam	male	2000
2	B	christine	female	2300
3	A	harry	male	2100
4	C	peter	male	1900
5	A	vivi	female	2400
6	B	Minnie	female	3000
7	A	Margaret	female	2100

Please load the dataset from the salary.csv file, create a new pandas Series with index as department name and values as the average salary of that department.

```
salary_info = pd.read_csv('salary.csv')
departments = ['A','B','C']
mean_salary_vec = []
for idep in departments:
    dep_salary_info = salary_info.loc[salary_info['department']==idep,'salary']
    avg_salary = np.mean(dep_salary_info)
    mean_salary_vec.append(avg_salary)
```

```
mean_salary_series = pd.Series(mean_salary_vec,index = departments)
```

8. Write Pandas codes to
    a. Create a 3*4 pandas DataFrame with random normal distributed values, and the index and columns are ['x','y','z'] and ['a','b','c','d'] respectively;
    b. Select the row having the smallest mean value to create a new pandas DataFrame;
    c. Select the column having the largest "mean positive value" to create a new pandas DataFrame. The "mean positive value" is calculated only over positive elements, and is zero if there is no positive element. For instance, the mean positive value is 3 for [2,3,4,0,-2], and is 0 for [-1,-2,-3].

```
import pandas as pd
import numpy as np

#a
df = pd.DataFrame(np.random.random((3, 4)),index=list('xyz'),columns=list('abcd'))

#b
df_row_mean = np.mean(df,axis=1)

#all are correct
df_min_row1 = df.loc[df_row_mean == np.min(df_row_mean),:]
df_min_row2 = df.loc[df_row_mean == np.min(df_row_mean)]
df_min_row3 = df[df_row_mean == np.min(df_row_mean)]

#c
mean_val_vec = []
for i_col in list('abcd'):
    df_col = df[i_col]
    positive_df = df_col[df_col>0]
    if len(positive_df):
        mean_val_vec.append(np.mean(positive_df))
    else:
        mean_val_vec.append(0)
mean_val_vec = np.array(mean_val_vec)
df_min_col = df.loc[:,mean_val_vec==np.max(mean_val_vec)]
```