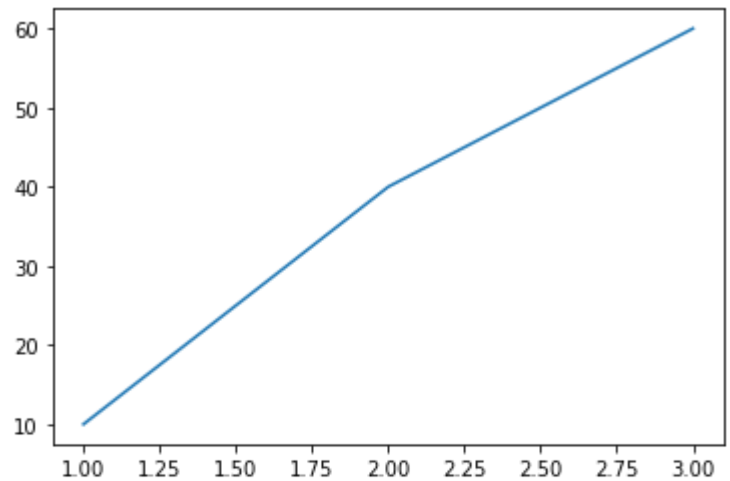
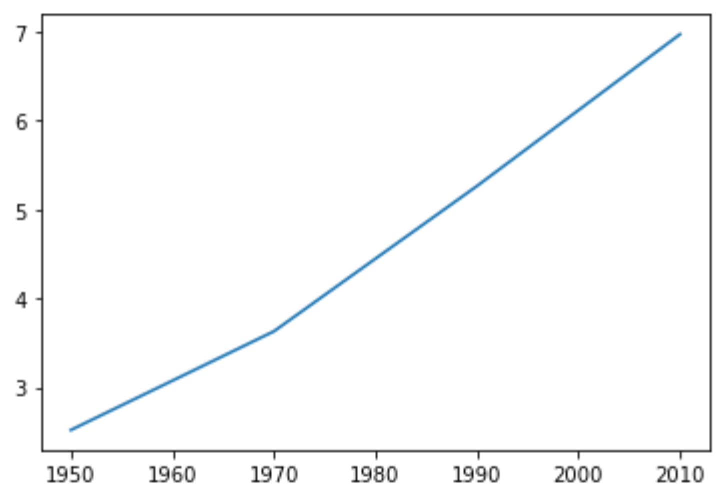
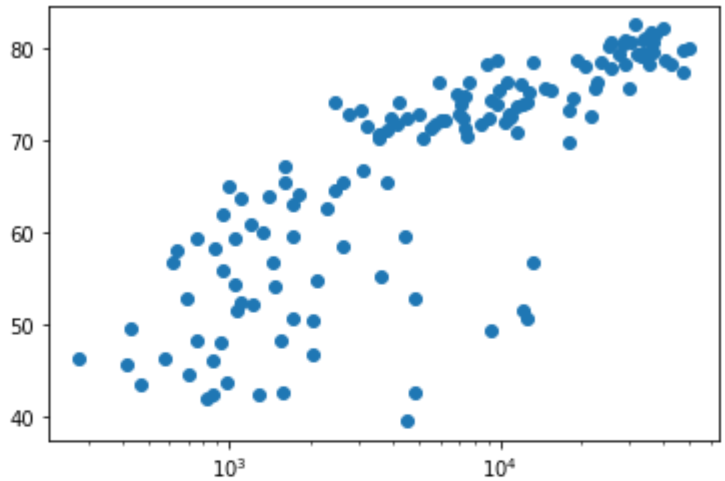
In [9]:
```
import matplotlib.pyplot as plt
%matplotlib inline

x = [1,2,3]
y = [10,40,60]
plt.plot(x,y)
plt.show()
```



In [13]:
```
import matplotlib.pyplot as plt
%matplotlib inline
year = [1950,1970,1990,2010]
pop = [2.519, 3.629, 5.263, 6.97]
plt.plot(year, pop)
plt.show()
```



In [21]:
```
gdp_cap = [974.5803384, 5937.029525999999, 6223.367465, 4797.231267, 12779

life_exp = [43.828, 76.423, 72.301, 42.731, 75.32, 81.235, 79.829, 75.635,

pop = [31.889923, 3.600523, 33.333216, 12.420476, 40.301927, 20.434176, 8.

col = ['red', 'green', 'blue', 'blue', 'yellow', 'black', 'green', 'red',


import matplotlib.pyplot as plt
%matplotlib inline
plt.scatter(gdp_cap, life_exp)
plt.xscale('log')
plt.show()
```

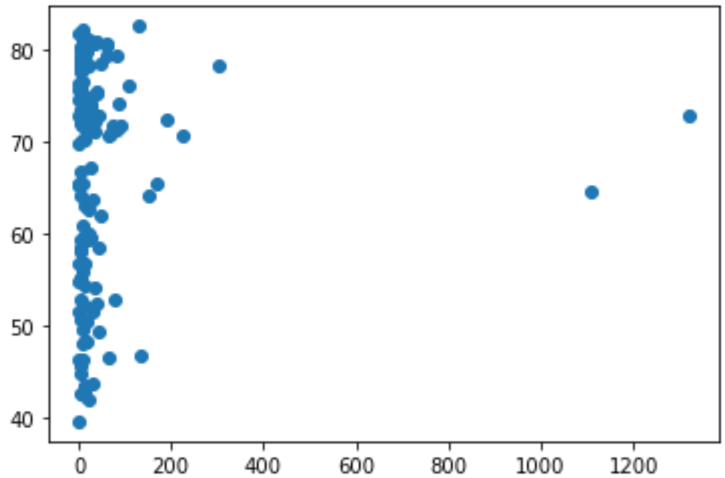


Ex1

ScatterPlot

In [28]:

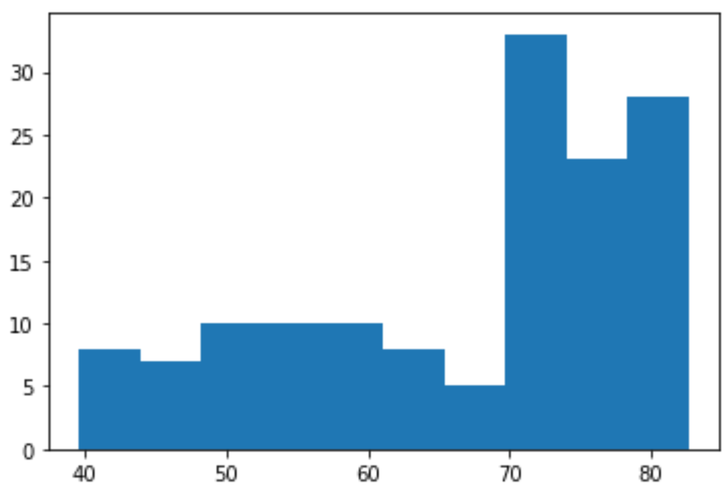
```
# gdp_cap = [974.5803384, 5937.029525999999, 6223.367465, 4797.231267, 127

life_exp = [43.828, 76.423, 72.301, 42.731, 75.32, 81.235, 79.829, 75.635,

pop = [31.889923, 3.600523, 33.333216, 12.420476, 40.301927, 20.434176, 8.



import matplotlib.pyplot as plt
%matplotlib inline
plt.scatter(pop, life_exp)
#plt.xscale('log')
plt.show()
```



Histogram of life_exp

In [31]:

```
plt.hist(life_exp)
plt.show()
```

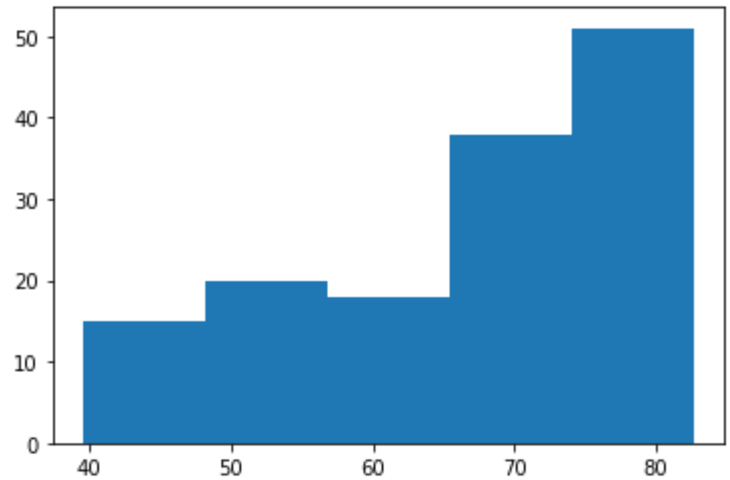


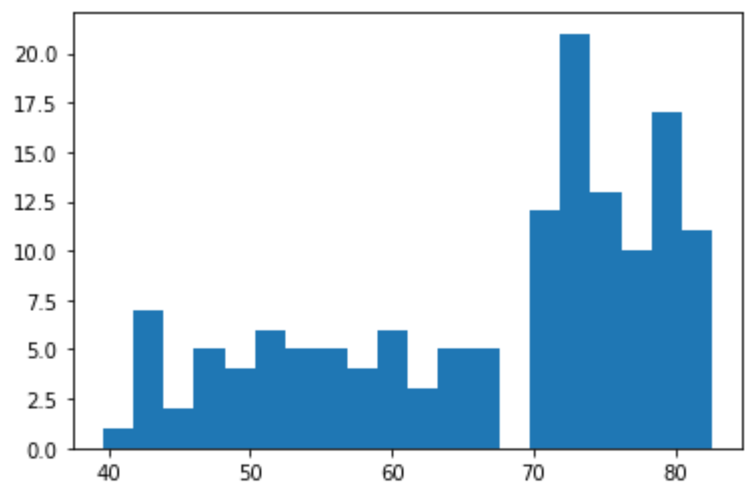
Ex2

Histogram of Life_exp

In [40]:

```
plt.hist(life_exp, bins = 5)
plt.show()


plt.hist(life_exp, bins = 20)
plt.show()
```



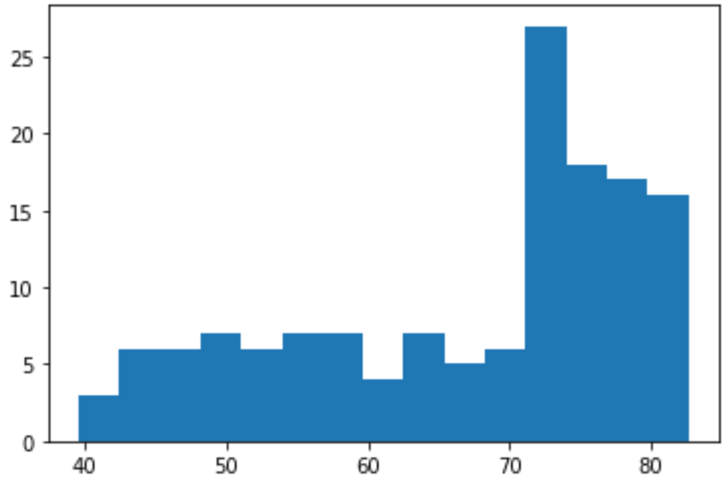


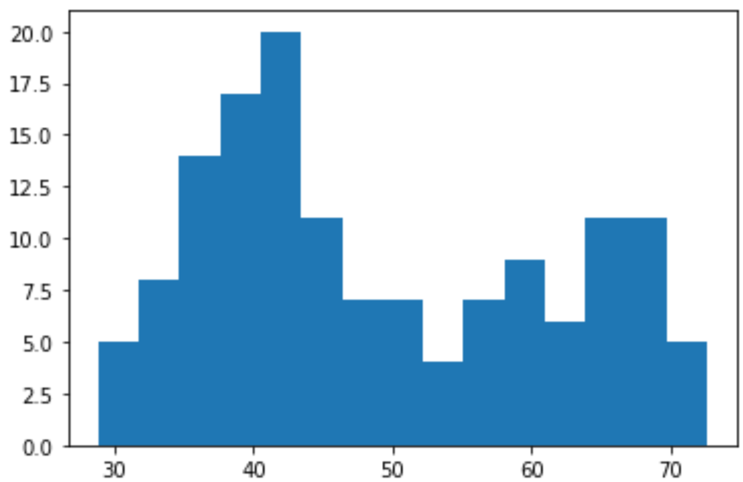
the bins 5 historgram is not easy to understand.

this bins 20 histogram is better because we can clearly see which timeline has more or less than the other.

Ex3

In [45]:
```
life_exp1950 = [28.8, 55.23, 43.08, 30.02, 62.48, 69.12, 66.8, 50.94, 37.4

plt.hist(life_exp, bins = 15)
plt.show()

plt.hist(life_exp1950, bins = 15)
plt.show()
```





I observe that they have differnt frequency even if the bins is the same.

life_exp 2007 has the highest frequency.

Ex4

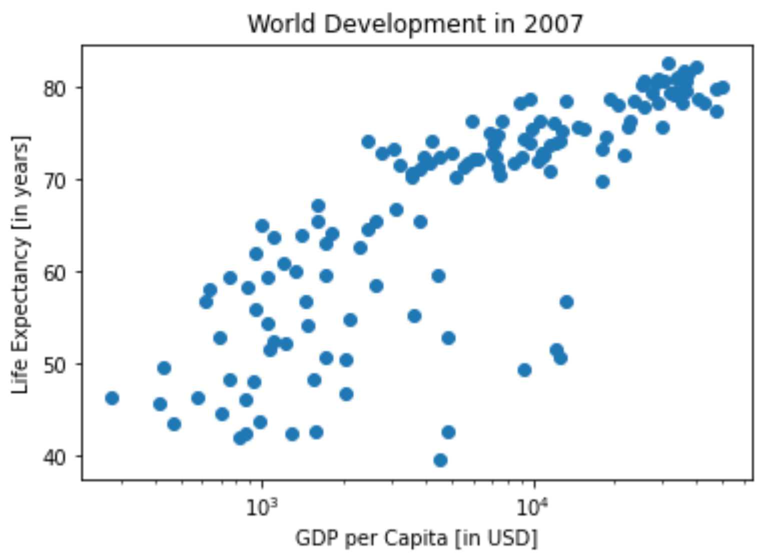
I will use Histogram because the grades on your exam follow a particular distribution

Ex5

I will use ScatterPlot because longer answers on exam questions lead to higher grades

Ex6

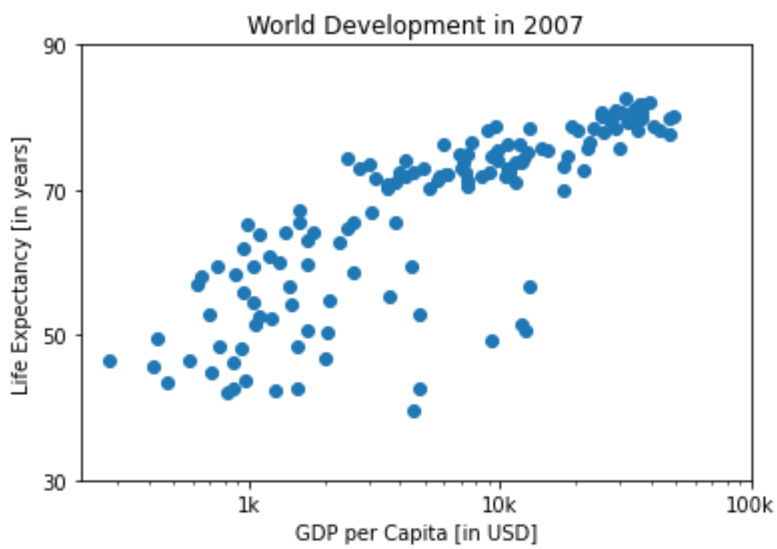
In [51]:
```
plt.scatter(gdp_cap, life_exp)
plt.xscale('log')

xlab = 'GDP per Capita [in USD]'
ylab = 'Life Expectancy [in years]'
title = 'World Development in 2007'

plt.xlabel(xlab)
plt.ylabel(ylab)
plt.title(title)
plt.show()
```



Ex7

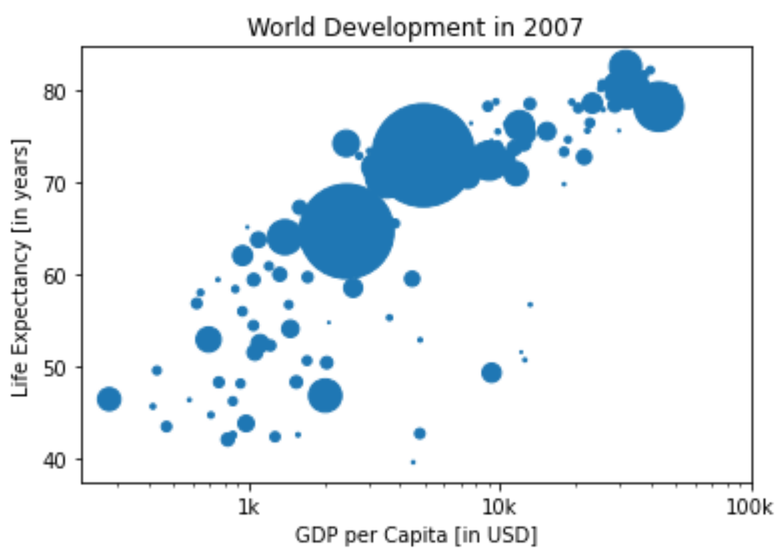
In [54]:

```
# Scatter plot
plt.scatter(gdp_cap, life_exp)
# Previous customizations
plt.xscale('log')
plt.xlabel('GDP per Capita [in USD]')
plt.ylabel('Life Expectancy [in years]')
plt.title('World Development in 2007')

# Definition of tick_val and tick_lab
tick_val = [1000,10000,100000]
tick_lab = ['1k','10k','100k']

tick_yval = [30, 50, 70 , 90]
tick_ylab = ['30', '50', '70', '90']

# Adapt the ticks on the x-axis
plt.xticks(tick_val,tick_lab)
plt.yticks(tick_yval, tick_ylab)

# After customizing, display the plot
plt.show()
```



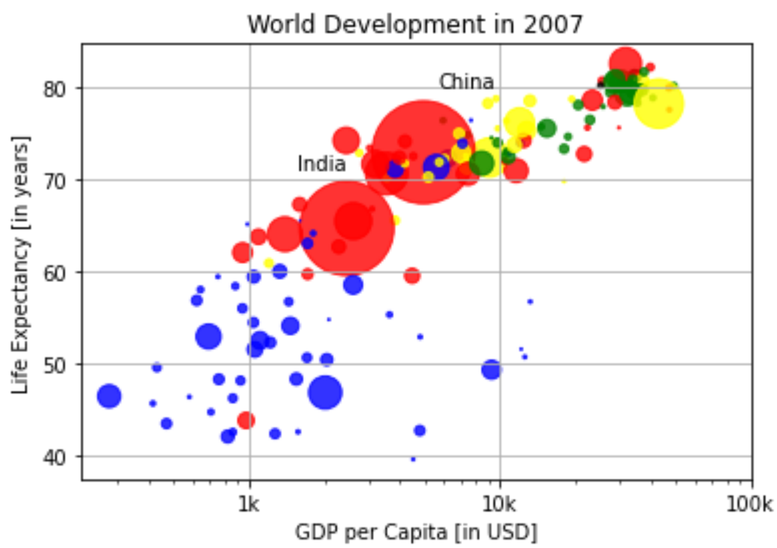
In [59]:

```
# Import numpy as np
import numpy as np

# Store pop as a numpy array: np_pop
np_pop = np.array(pop)

# Double np_pop
np_pop = np_pop * 2

# Update: set s argument to np_pop
plt.scatter(gdp_cap, life_exp, s = np_pop)

# Previous customizations
plt.xscale('log')
plt.xlabel('GDP per Capita [in USD]')
plt.ylabel('Life Expectancy [in years]')
plt.title('World Development in 2007')
plt.xticks([1000, 10000, 100000],['1k', '10k', '100k'])

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



In [68]:

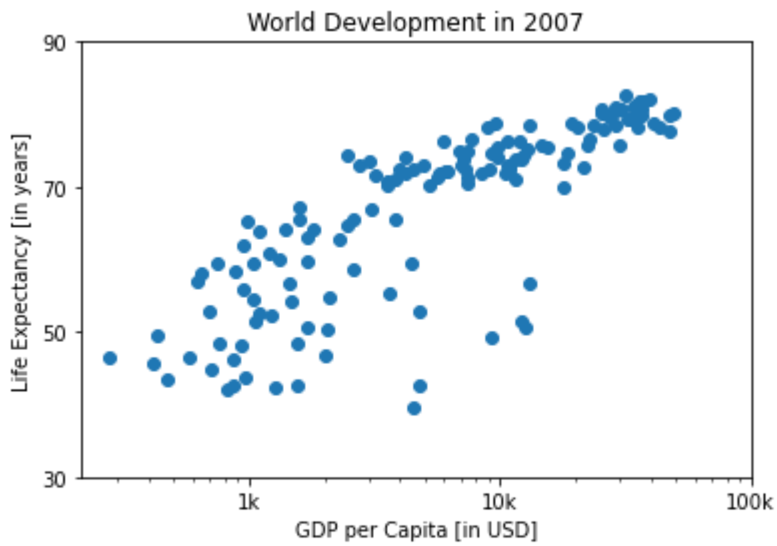
```
col = ['red', 'green', 'blue', 'blue', 'yellow', 'black', 'green', 'red',

# Scatter plot
plt.scatter(x = gdp_cap, y = life_exp, s = np.array(pop) * 2, c = col, alpl
# Previous customizations
plt.xscale('log')
plt.xlabel('GDP per Capita [in USD]')
plt.ylabel('Life Expectancy [in years]')
plt.title('World Development in 2007')
plt.xticks([1000,10000,100000], ['1k','10k','100k'])
# Additional customizations
plt.text(1550, 71, 'India')
plt.text(5700, 80, 'China')
# Add grid() call
plt.grid(1)
# Show the plot
plt.show()
```

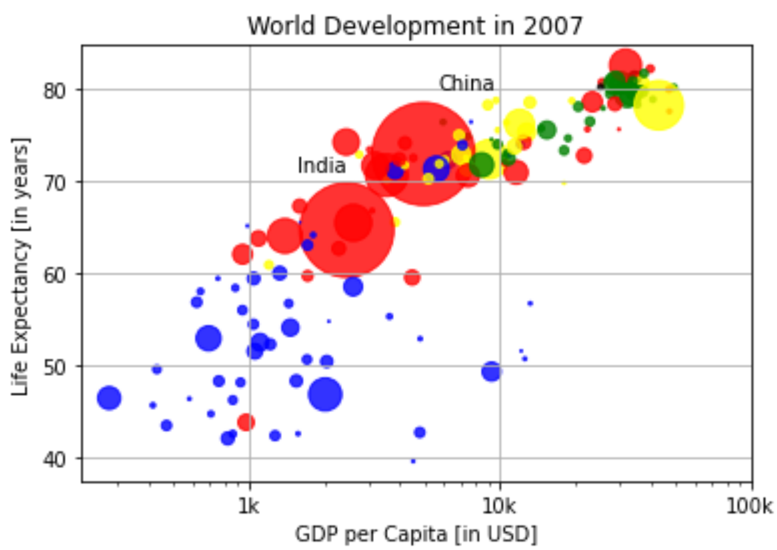


Ex8

In [70]:

```
# Scatter plot
plt.scatter(gdp_cap, life_exp)
# Previous customizations
plt.xscale('log')
plt.xlabel('GDP per Capita [in USD]')
plt.ylabel('Life Expectancy [in years]')
plt.title('World Development in 2007')

# Definition of tick_val and tick_lab
tick_val = [1000,10000,100000]
tick_lab = ['1k','10k','100k']

tick_yval = [30, 50, 70 , 90]
tick_ylab = ['30', '50', '70', '90']

# Adapt the ticks on the x-axis
plt.xticks(tick_val,tick_lab)
plt.yticks(tick_yval, tick_ylab)

# After customizing, display the plot
plt.show()


col = ['red', 'green', 'blue', 'blue', 'yellow', 'black', 'green', 'red',

# Scatter plot
plt.scatter(x = gdp_cap, y = life_exp, s = np.array(pop) * 2, c = col, alpl
# Previous customizations
plt.xscale('log')
plt.xlabel('GDP per Capita [in USD]')
plt.ylabel('Life Expectancy [in years]')
plt.title('World Development in 2007')
plt.xticks([1000,10000,100000], ['1k','10k','100k'])
# Additional customizations
plt.text(1550, 71, 'India')
plt.text(5700, 80, 'China')
# Add grid() call
plt.grid(1)
# Show the plot
plt.show()
```





the first one is true. (o The countries in blue, corresponding to Africa, have both low life expectancy and a low GDP per capita.)

In [72]:

```

```

In [ ]:

```

```