In [3]:

```
import numpy as np
nations = np.genfromtxt("nations.csv", delimiter=",", skip_header=True)
```

In [4]:

```
# How many people live on earth?

amtPeopleOnEarth = np.sum(nations, axis=0)[6]
print(amtPeopleOnEarth)
```

```
6482276104.0
```

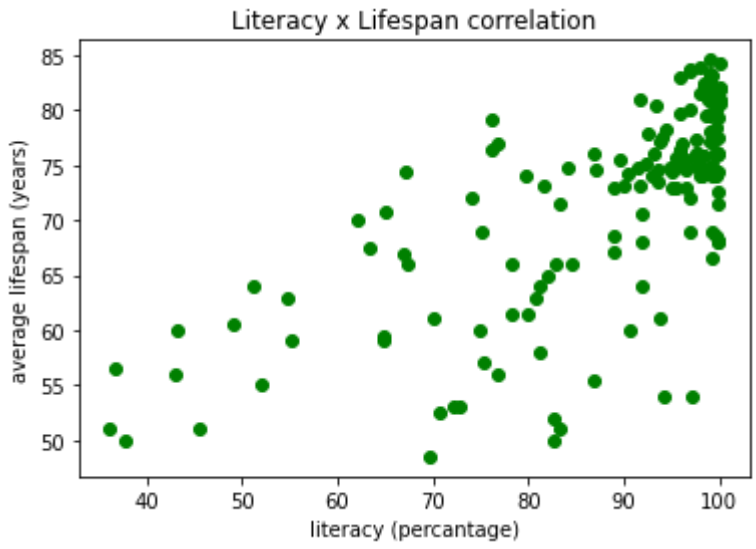
In [5]:

```
# What is the average life expectancy of the world population?

avgLifeExp = np.mean(nations, axis=0)[4]
print(avgLifeExp)
```

```
71.69006134969325
```

In [6]:

```
# Do people who can read live longer? Back up your claim with a diagram.

import matplotlib.pyplot as plt
plt.title("Literacy x Lifespan correlation")
plt.ylabel("average lifespan (years)")
plt.xlabel("literacy (percantage)")


plt.plot(nations[:,5]*100, nations[:,4], "og")

plt.show()
```



There appears to be a correlation between high literacy and a long life expactency. Although, high literacy does not always come with a high average lifespan. But on the opposite side of the spectrum there are no low-

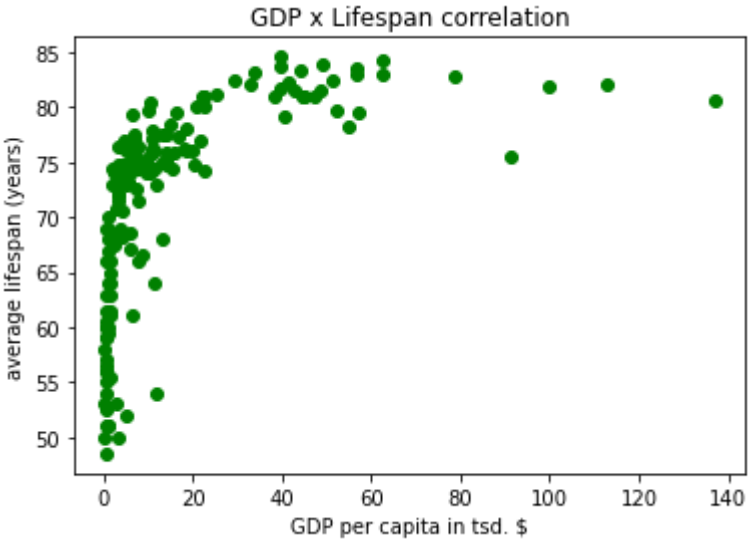
lifespan countries with higher-end literacy. In conclusion: A high literacy is essential but not sufficient for long average lifespans.

In [27]:

```
# Plot the average life expectancy in each country (Y-axis) as a function of GDP per capita

perCapita = nations[:,3]/nations[:,6]*1000

plt.title("GDP x Lifespan correlation")
plt.ylabel("average lifespan (years)")
plt.xlabel("GDP per capita in tsd. $")

plt.plot(perCapita, nations[:,4], "og")
```

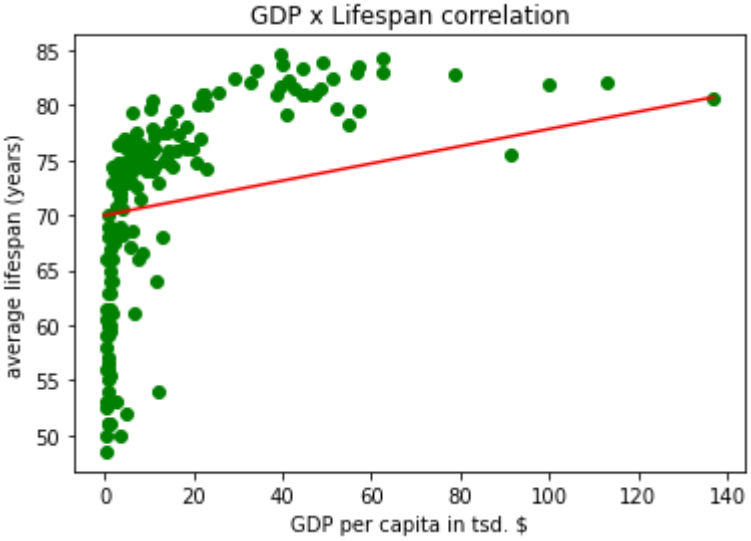
Out[27]:

```
[<matplotlib.lines.Line2D at 0x22523f72d00>]
```



In [28]:

```
# Additionally, draw a straight line in the same plot, which approximately reflects the cou

plt.title("GDP x Lifespan correlation")
plt.ylabel("average lifespan (years)")            # <--- same as before, since plot.show()
plt.xlabel("GDP per capita in tsd. $")

plt.plot(perCapita, nations[:,4], "og")

x, y = [0, 137], [70, 80.75] # rough approximation
plt.plot(x, y, color= "r")
plt.show()
```



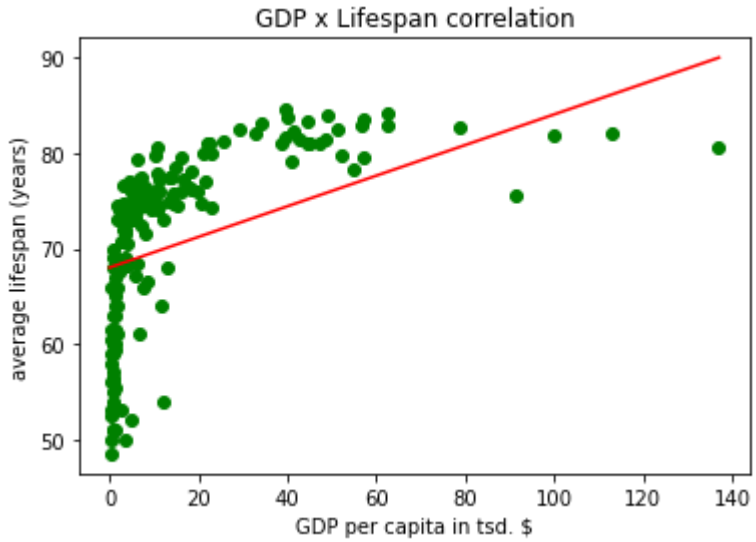
In [32]:

```
# Create the appropriate linear equation and store it in a function.

def calcLifeSpan(gdpPerCap): # takes GDP per capita in tsd $ as parameter

    m=(70-80.75)/(0-137) # numbers from above (change in height)
    n=70-0*m # numbers from above (shift on y axis)

    # final equation: y = 0.0766*x + 70

    return gdpPerCap*m+n



# Calculate how well the line approximates the data points by calculating the root-mean-squ

import math

sum = 0
spans = nations[:,4] # renamed lifespan array

for x in range(len(perCapita)): # for every data point
    sum += (calcLifeSpan(perCapita[x])-spans[x])**2 # sum up squares of difference between

rmse = math.sqrt(sum/len(perCapita)) # root the sum divided by amount of sum-ups

print("RMSE: ", rmse) # result
```

```
RMSE:  8.172791904478894
```

In [38]:

```
# Change the straight line by hand so that the error or deviation is smaller. Make a note o

plt.title("GDP x Lifespan correlation")
plt.ylabel("average lifespan (years)")            # <--- same as before, since plot.show()
plt.xlabel("GDP per capita in tsd. $")

plt.plot(perCapita, nations[:,4], "og")

x, y = [0, 137], [68, 90] # rough approximation
plt.plot(x, y, color= "r")
plt.show()


def calcLifeSpan(gdpPerCap): # takes GDP per capita in tsd $ as parameter

    m=(68-90)/(0-137) # numbers from above (change in height)
    n=68-0*m # numbers from above (shift on y axis)

    # final equation: y = 0.1605*x + 68

    return gdpPerCap*m+n




sum = 0
spans = nations[:,4] # renamed lifespan array

for x in range(len(perCapita)): # for every data point
    sum += (calcLifeSpan(perCapita[x])-spans[x])**2 # sum up squares of difference between

rmse = math.sqrt(sum/len(perCapita)) # root the sum divided by amount of sum-ups

print("RMSE: ", rmse) # result
```




```
RMSE:  7.688873373507815
```

```
After trying for several minutes, I managed to achieve a slightly better result:

FINAL EQUATION:
```

```
   y = 0.1605*x + 68    |    RMSE    ~ 7.688873373507815

PREVIOUSLY:
   y = 0.0766*x + 70    |    RMSE    ~ 8.172791904478894
```