# Afleveringsopgave 3

In [331...

```
import numpy as np
```

In [332...

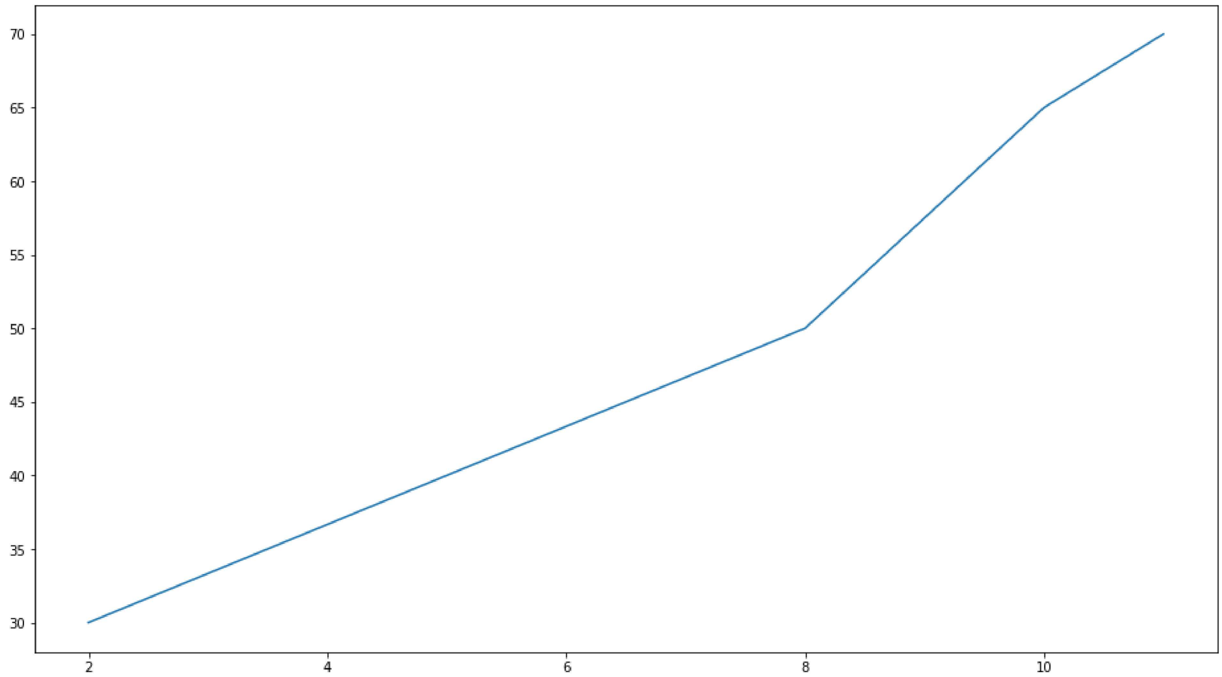
```
import matplotlib.pyplot as plt
```

a)

In [333...

```
tid = [2.0, 5.0, 8.0, 10.0, 11.0]
temperatur = [30.0, 40.0, 50.0, 65.0, 70.0]

fig, ax = plt.subplots()
ax.plot(tid, temperatur)
plt.show()
print("\n")
```




b)

Vi har en andengradsligning:

p(x) = a + b *x* + *c* x**2

Med den kan opstille et ligningssytem, hvor a, b og c er de eneste ubekendte:

50.0 = a + 8b + 64c

65.0 = a + 10b + 100c

70.0 = a + 11b + 121

In [334...

```
A = np.array([[1.0, 8.0, 64.0],
              [1.0, 10.0, 100.0],
              [1.0, 11.0, 121.0]])
```

```python
B = np.array([50.0, 65.0, 70.0])[:, np.newaxis]

C = np.hstack([A,B])
C
```

Out[334…
```
array([[  1.,   8.,  64.,  50.],
       [  1.,  10., 100.,  65.],
       [  1.,  11., 121.,  70.]])
```

In [335…
```python
C[[1,2],:] -= C[0,:]        # Først trækker vi række 0 fra 1    #
                            #    og 2 for at få 0 i søjle 0     #
C[2,:] -= 3/2 * C[1,:]      # Nu trækker vi en skaleret søjle   #
                            #    1 fra 2, for at få 0 i søjle 1 #
C[1,:] *= 1/2               # Til sidst bruger vi skalarer til  #
                            #    at etablere pivotelementerne   #
C[2,:] *= 1/3
C
```

Out[335…
```
array([[ 1.    ,  8.    , 64.    , 50.    ],
       [ 0.    ,  1.    , 18.    ,  7.5   ],
       [ 0.    ,  0.    ,  1.    , -0.8333]])
```

In [336…
```python
C[1,:] -= 18 * C[2,:]               # Nu reducerer vi koefficientmatricen til en
C[0,:] -= 64 * C[2,:] + 8 * C[1,:]  # indentitetsmatrix for at finde løsningen
C
```

Out[336…
```
array([[  1.    ,   0.    ,   0.    , -76.6667],
       [  0.    ,   1.    ,   0.    ,  22.5   ],
       [  0.    ,   0.    ,   1.    ,  -0.8333]])
```

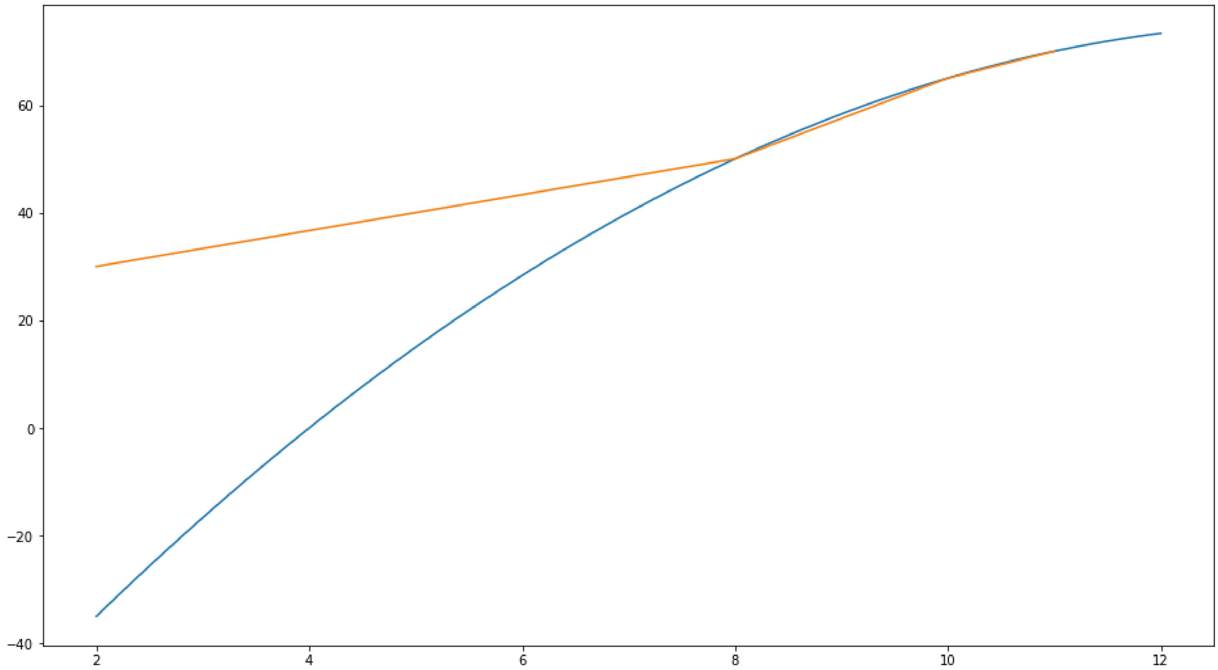
In [337…
```python
a, b, c = C[:,3]
a, b, c #Løsningerne på ligningssystemet
```

Out[337…
```
(-76.66666666666667, 22.5, -0.8333333333333333)
```

In [338…
```python
x = np.linspace(2, 12, 200)

y = a + b * x + c * x**2

fig, ax = plt.subplots() # Tegning
ax.plot(x, y)
ax.plot(tid, temperatur)
plt.show()
```




Her kan vi se, at polynomiet passer rimelig godt på vores data i intervallet mellem vores tre punkter. Bevæger vi os ud herfra, afviger funktionen dog hurtigt fra vores data.

**c)**

Hvis vi skal have et polynomium, der går igennem fem punkter, er vi nødt til at have fem variable. Det vil sige, at vi som minium skal bruge et fjerdegradspolynomium:

p(x) = a + bx + cx^2 + dx^3 + ex^4

In [339…

```
A = np.array([[1.0, 2.0, 4.0, 8.0, 16.0],
              [1.0, 5.0, 25.0, 125.0, 625.0],
              [1.0, 8.0, 64.0, 512.0, 4096.0],
              [1.0, 10.0, 100.0, 1000.0, 10000.0],
              [1.0, 11.0, 121.0, 1331.0, 14641.0]])

B = np.array([35.0, 40.0, 50.0, 65.0, 70.0])[:, np.newaxis]

a, b, c, d, e = np.linalg.solve(A,B)
a, b, c, d, e
```

Out[339…

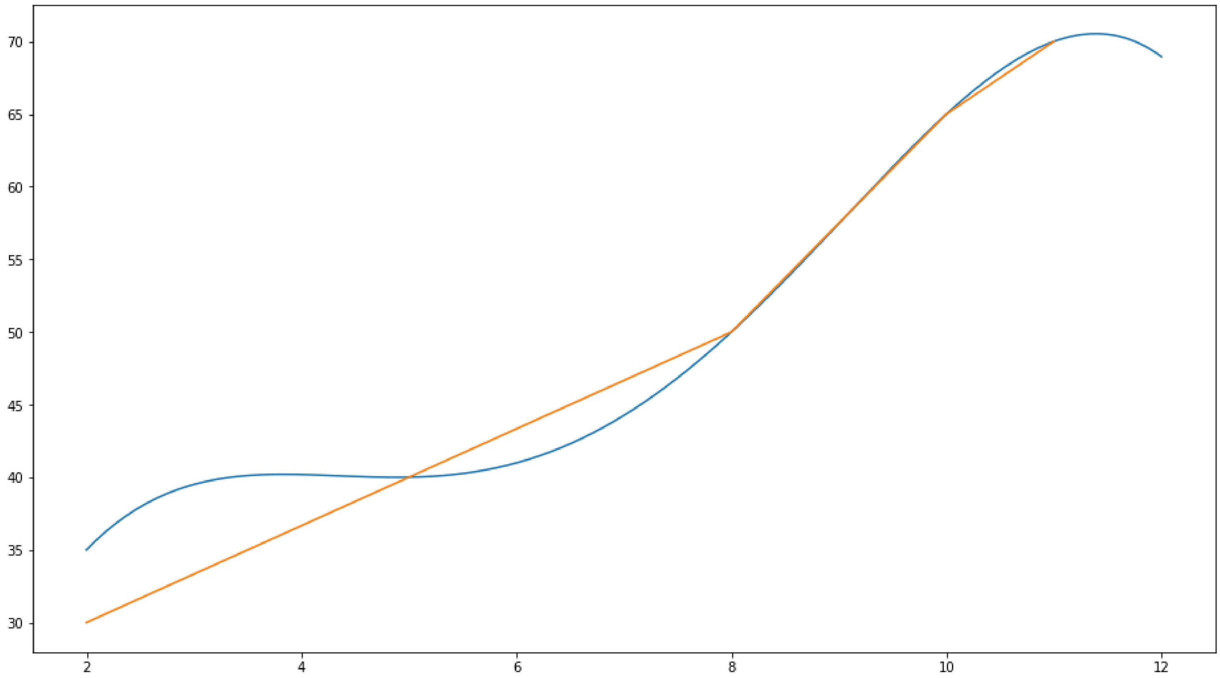
```
(array([-1.9753]),
 array([32.8549]),
 array([-9.0972]),
 array([1.034]),
 array([-0.0386]))
```

In [340…

```
x = np.linspace(2, 12, 200)

y = a + b * x + c * x**2 + d * x**3 + e * x**4

fig, ax = plt.subplots() # Tegning
ax.plot(x, y)
ax.plot(tid, temperatur)
plt.show()
```


70
65
60
55
50
45
40
35
30
2
4
6
8
10
12