Tall 1)

a) $\sigma = \sqrt{\frac{1}{N}\sum(x-\mu)^2}$, $\mu = \frac{\sum x_i}{N} =$

Bruker Python:

Script

```python
#Excersise 1 for TTT4280

import numpy as np
import matplotlib.pyplot as plt

#1.1
#Find the mean value and standard deviation of the data set
data = np.loadtxt('Øving 1/data.txt', delimiter=',')
meanDataValue = data.mean()
stdDataValue = data.std()
print("Mean value: ", meanDataValue)
print("Standard deviation: ", stdDataValue)

```

Result

```
Eier@MacBook-Air-2 Sensorer og intrumentering % /usr/
crypy.py"
Mean value:  20.485
Standard deviation:  0.10618380290797719
Eier@MacBook-Air-2 Sensorer og intrumentering %
```

b) $CI = \bar{X} + t_\mu \frac{S}{\sqrt{n}}$, $t_\mu = 2,093$

$CI = 20,485 \pm 2.093 \cdot \frac{0,106}{\sqrt{20}} = \underline{[20,435 , 20.534]}$

c) $T \in \mu_T \pm z\sigma\sqrt{1+\frac{1}{n}} = \underline{[20.258, 20,711]}$

~~Task 2)~~

a) Use same code as before with new values:

Result:

```
crypy.py"
Mean value:  20.330000000000002
Standard deviation:  0.095393920141694
Eier@MacBook-Air-2 Sensorer og intrumentering %
```

$$CI = \bar{x} \pm t_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}} = 20.33 \pm 2.093 \cdot \frac{0.095}{\sqrt{20}} = [20.285, 20.374]$$

b) No the two CI's are not overlapping. This suggest that there is a statistically significant difference between the mean's measured by Olav and Ove.

We can also see that OVC's system is closest to the ideal temp 20° and has a smaller standard deviation.

c) Only using the first value.

New code:

```
11  #1.2c
12  #Caclculate for half the data set
13  halfData = data[:int(len(data)/2)]
14  meanHalfDataValue = halfData.mean()
15  stdHalfDataValue = halfData.std()
16  print("Mean value half: ", meanHalfDataValue)
17  print("Standard deviation half: ", stdHalfDataValue)
18
```

Ove's results

```
Eier@MacBook-Air-2 Sensorer og intrumentering %
Mean value half:  20.380000000000003
Standard deviation half:  0.07483314773547865
Eier@MacBook-Air-2 Sensorer og intrumentering %
```

Olav's result:

```
Eier@MacBook-Air-2 Sensorer og intrumentering %
Mean value half:  20.509999999999998
Standard deviation half:  0.12206555615733793
```

Ore's new CI:

$$CI: \bar{x} + t_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}, \quad t_{\alpha/2} = 2,262$$

$$: 20.380 \pm 2.262 \cdot \frac{0.07483}{\sqrt{10}} = [20.326, 20,433]$$

Olav's new CI:

$$CI: \bar{x} + t_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}$$

$$20.510 \pm 2.262 \frac{0.1220}{\sqrt{10}} = [20.422, 20.597]$$

Now they overlap!

Task 3)

a) max: $10k\Omega + 1\%\,10k\Omega = \underline{10.1k\Omega}$

min: $10k\Omega - 1\%\,10\Omega = \underline{99.9k\Omega}$

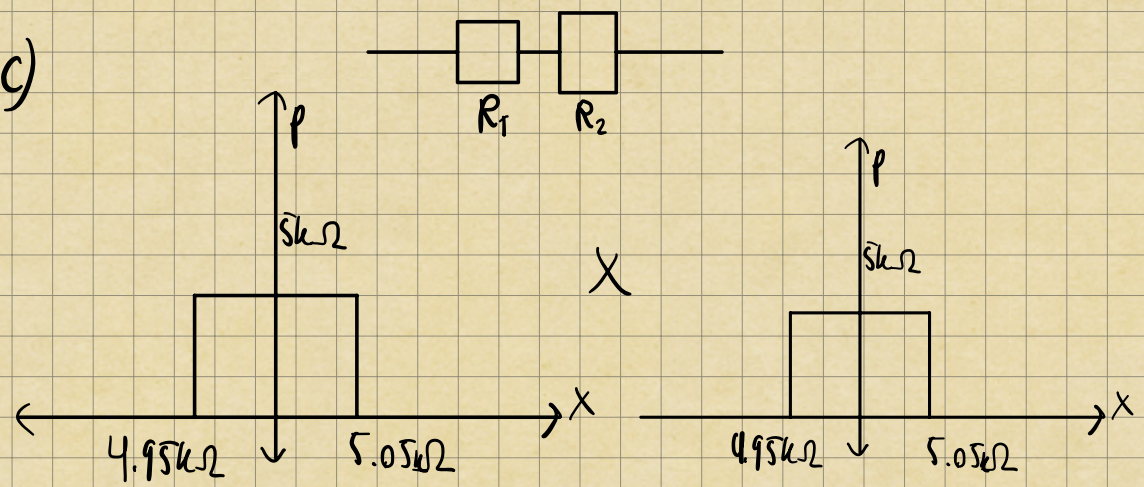
Standard deviation for uniform distribution:

$$\sigma = \sqrt{\frac{(max - min)^2}{12}} = \underline{57.7\Omega}$$

b)

$$RSD = \frac{\sigma}{|m|} \cdot 100\% = \frac{57.7\Omega}{10 \cdot 10^3\Omega} \cdot 100\% = \underline{0.577\%}$$

c)



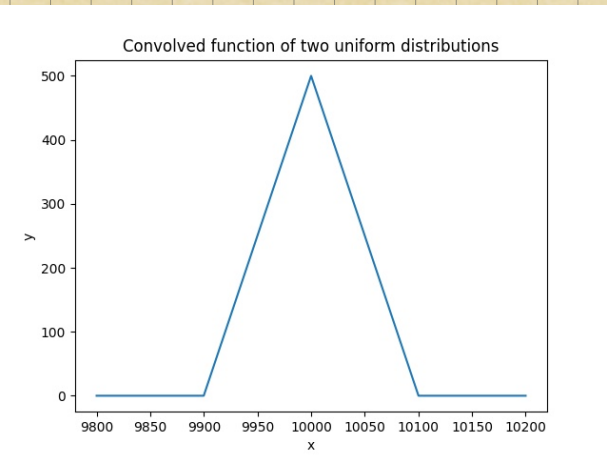
$$g(x) = f(x) * f(x)$$

This is straw so I do it in python.

Code:

```
#1.3c

# Example usage
x_values = np.linspace(4900, 5100, 1000)  # Generate 1000 points from 4900 to 5100
y_values = continuous_uniform_distribution(x_values)

# Convolve the distribution with itself
convolved = np.convolve(y_values, y_values, mode='full')

# Generate new x values for the convolved function
convolved_x_values = np.linspace(2*x_values[0], 2*x_values[-1], len(convolved))

# Plot the convolved function
plt.plot(convolved_x_values, convolved)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Convolved function of two uniform distributions')
plt.show()
```

Result



d) $Var(X+Y) = Var(X) + Var(Y)$

```
#1.4c
rng = default_rng()
delta_r = 100

resistor = 5000
deviationToResitor = resistor * 0.01

totaleValues = 6000

Rvalue1 = resistor +2*deviationToResitor*(rng.random(totaleValues)-0.5)
Rvalue2 = resistor +2*deviationToResitor*(rng.random(totaleValues)-0.5)

r_values = Rvalue1 + Rvalue2
#Find the relative standard deviation
RelativeStandardDeviation = r_values.std()/r_values.mean()*100
print("Relative standard deviation: ", RelativeStandardDeviation)
```

```
branch 'main' set up to track 'origin/main'.
Eier@dhcp-10-24-20-139 Sensorer og intrumentering % /usr/bin/
Elsys/Sensorer og intrumentering/Øving 1/scrypy.py"
Relative standard deviation:  0.41237072246253687
```