1) $$\sum_{i,j,k} M_{ik} O_{kj} N_{jl} V_i =$$

$$= N_{lj}^T O_{jk}^T M_{ki}^T V_i$$

$$= V_i^T M_{ik} O_{kj} N_{jl}$$

$$(W^T B C^T A^T u)^T = u^T A C B^T W$$

2) $$x^T A x$$

$$\nabla_x (x^T A x) = \frac{\partial}{\partial x_i}\left(\sum_k x^T_j A_{jk} x_k\right)$$

$$= \frac{\partial}{\partial x_i}(x_j A_{jk} x_k)$$

$$= A_{jk} \frac{\partial x_j}{\partial x_i} x_k + A_{jk} \frac{\partial x_k}{\partial x_i} x_j$$

$$= A_{jk} \delta_{ji} x_k + A_{jk} \delta_{ki} x_j$$

$$= x_k A_{kj}^T \delta_{ji} + x_j A_{jk} \delta_{ki}$$

<u>If Symmetric</u>

$$2A(x_k + x_j)$$

```
from scipy.stats import norm
import numpy as np
import pandas as pd
from datetime import datetime
import pandas_datareader.data as web
import matplotlib.pyplot as plt
import math
import random

#Question 3
amd = web.DataReader("AMD", "yahoo", datetime(2016,1,1), datetime(2021,1,1))
t = amd.index
n = len(t) - 1
stockPrice = amd["Adj Close"]

temp = stockPrice.pct_change()
dailyReturns = np.array(temp)
dailyReturns.sort()

top = int(n*0.01)
print("1st percentile value:", dailyReturns[len(dailyReturns) - 1 - top])

temp = []
for i in dailyReturns:
  if i > 0.05:
    temp.append(i)

meanReturn = (np.sum(temp))/(len(temp))
print("Mean of all returns greater than 5%:", meanReturn)

print("The lowest return is", (np.min(temp) - meanReturn)/(np.std(temp)),
      "sigma away from the mean")
```

```
1st percentile value: 0.11877402055862474
Mean of all returns greater than 5%: 0.08443754387706268
The lowest return is -0.6573236356121318 sigma away from the mean
```