Question 1a

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [ ]:

```
def fun(x):
  f=x[0]*np.log(x[0])+x[1]*np.log(x[1])+x[2]*np.log(x[2])+x[3]*np.log(x[3])+x[4]*np.log(x[4])
  return f
```

In [ ]:

```
def gradf(x):
  g=np.array([1+np.log(x[0]),
              1+np.log(x[1]),
              1+np.log(x[2]),
              1+np.log(x[3]),
              1+np.log(x[4])
              ])
  return g
```

In [ ]:

```
def Hessian(x):
  g=np.array([[1/x[0],0,0,0,0],
              [0,1/x[1],0,0,0],
              [0,0,1/x[2],0,0],
              [0,0,0,1/x[3],0],
              [0,0,0,0,1/x[4]]
             ])
  return g
```

In [ ]:

```
x0=np.array([1/3,1/4,1/6,1/6,1/12])
iter=0
print("x",iter," : ",x0)
print("g",iter," : ",gradf(x0))
```

```
x 0  :  [0.33333333 0.25       0.16666667 0.16666667 0.08333333]
g 0  :  [-0.09861229 -0.38629436 -0.79175947 -0.79175947 -1.48490665]
```

In [ ]:

```
A=np.array([[1,1,1,1,1]])
H1=np.column_stack((Hessian(x0),A.T))
H2=np.column_stack((A,[0]))
H=np.vstack((H1,H2))
print(H)
print(np.linalg.inv(H))
```

```
[[ 3.  0.  0.  0.  0.  1.]
 [ 0.  4.  0.  0.  0.  1.]
 [ 0.  0.  6.  0.  0.  1.]
 [ 0.  0.  0.  6.  0.  1.]
 [ 0.  0.  0.  0. 12.  1.]
 [ 1.  1.  1.  1.  1.  0.]]
[[ 0.22222222 -0.08333333 -0.05555556 -0.05555556 -0.02777778  0.33333333]
 [-0.08333333  0.1875     -0.04166667 -0.04166667 -0.02083333  0.25      ]
 [-0.05555556 -0.04166667  0.13888889 -0.02777778 -0.01388889  0.16666667]
 [-0.05555556 -0.04166667 -0.02777778  0.13888889 -0.01388889  0.16666667]
 [-0.02777778 -0.02083333 -0.01388889 -0.01388889  0.07638889  0.08333333]
 [ 0.33333333  0.25        0.16666667  0.16666667  0.08333333 -1.        ]]
```

In [ ]:

```
t=np.vstack((-gradf(x0).reshape(-1,1),[0]))
t
```

```
Out[ ]: array([[0.09861229],
               [0.38629436],
               [0.79175947],
               [0.79175947],
               [1.48490665],
               [0.        ]])
```

```
In [ ]:  d_mu=np.dot(np.linalg.inv(H),t)
         d_mu
```

```
Out[ ]: array([[-0.13949804],
               [-0.03270301],
               [ 0.04577551],
               [ 0.04577551],
               [ 0.08065002],
               [ 0.5171064 ]])
```

```
In [ ]:  d_mu.reshape(1,-1)[0][0:5]
```

```
Out[ ]: array([-0.13949804, -0.03270301,  0.04577551,  0.04577551,  0.08065002])
```

```
In [ ]:  print(np.dot(A.T,d_mu[5]))
         print(A.T.shape)
         print(d_mu[5].shape)
```

```
[0.5171064 0.5171064 0.5171064 0.5171064 0.5171064]
(5, 1)
(1,)
```

```
In [ ]:  while np.linalg.norm(gradf(x0)+np.dot(A.T,d_mu[5]))>0.001 and iter<100:

           x1=x0+d_mu.reshape(1,-1)[0][0:5]
           print("x",iter+1," : ",x1)
           print("f",iter+1," : ",fun(x1))
           print("g",iter+1," : ",gradf(x1))
           print("\n===========\n")

           x0,iter=x1,iter+1
           H1=np.column_stack((Hessian(x0),A.T))
           H2=np.column_stack((A,[0]))
           H=np.vstack((H1,H2))
           d_mu=np.dot(np.linalg.inv(H),np.vstack((-gradf(x0).reshape(-1,1),[0])))
         print("\n===========\n")
         print("x",iter," : ",x0)
         print("f",iter," : ",fun(x1))
         print("g",iter," : ",gradf(x0))
```

```
x 1  :  [0.1938353  0.21729699 0.21244218 0.21244218 0.16398335]
f 1  :  -1.6043988037453452
g 1  :  [-0.64074646 -0.52649024 -0.54908543 -0.54908543 -0.80799035]

===========

x 2  :  [0.20088076 0.20036769 0.20069128 0.20069128 0.19736898]
f 2  :  -1.6094158694703842
g 2  :  [-0.6050438  -0.60760113 -0.60598745 -0.60598745 -0.62268031]

===========

x 3  :  [0.20000249 0.20000408 0.20000323 0.20000323 0.19998697]
f 3  :  -1.60943791190025
g 3  :  [-0.60942545 -0.60941752 -0.60942176 -0.60942176 -0.60950307]

===========
===========

x 3  :  [0.20000249 0.20000408 0.20000323 0.20000323 0.19998697]
f 3  :  -1.60943791190025
```

Question 1b

In [ ]:

```
def fun(x):
  f=x[0]*np.exp(-x[0])+x[1]*np.exp(-x[1])+x[2]*np.exp(-x[2])+x[3]*np.exp(-x[3])
  return f
```

In [ ]:

```
def gradf(x):
  g=np.zeros(4)
  g=(1-x)*np.exp(-x)
  return g
```

In [ ]:

```
def Hessian(x):
  g=np.zeros(4)
  g=(x-2)*np.exp(-x)
  H=np.diag(g)
  return H
```

In [ ]:

```
print(fun(np.array([0,0,0,0])))
print(gradf(np.array([0,0,0,0])))
print(Hessian(np.array([0,0,0,0])))
```

```
0.0
[1. 1. 1. 1.]
[[-2.  0.  0.  0.]
 [ 0. -2.  0.  0.]
 [ 0.  0. -2.  0.]
 [ 0.  0.  0. -2.]]
```

In [ ]:

```
x0=np.array([2/3,1/3,0,0])
iter=0
print("x",iter," : ",x0)
print("g",iter," : ",gradf(x0))
```

```
x 0  :  [0.66666667 0.33333333 0.         0.        ]
g 0  :  [0.17113904 0.47768754 1.         1.        ]
```

In [ ]:

```
A=np.array([[1,1,1,1],[1,-2,3,-4]])
H1=np.column_stack((Hessian(x0),A.T))
H2=np.column_stack((A,[0,0],[0,0]))
H=np.vstack((H1,H2))
print(H)
print(np.linalg.inv(H))
```

```
[[-0.68455616  0.          0.          0.          1.          1.        ]
 [ 0.         -1.19421885  0.          0.          1.         -2.        ]
 [ 0.          0.         -2.          0.          1.          3.        ]
 [ 0.          0.          0.         -2.          1.         -4.        ]
 [ 1.          1.          1.          1.          0.          0.        ]
 [ 1.         -2.          3.         -4.          0.          0.        ]]
[[-0.62972845  0.21618474  0.38803896  0.02550474  0.46533431  0.10358121]
 [ 0.21618474 -0.49475636 -0.01305871  0.29163033  0.23504461 -0.08705401]
 [ 0.38803896 -0.01305871 -0.27343963 -0.10154062  0.17189132  0.09374314]
 [ 0.02550474  0.29163033 -0.10154062 -0.21559445  0.12772976 -0.11027034]
 [ 0.46533431  0.23504461  0.17189132  0.12772976  0.30592988  0.01261759]
 [ 0.10358121 -0.08705401  0.09374314 -0.11027034  0.01261759  0.05828957]]
```

In [ ]:

```
t=np.vstack((-gradf(x0).reshape(-1,1),[0],[0]))
t
```

Out[ ]:

```
array([[-0.17113904],
       [-0.47768754],
       [-1.        ],
       [-1.        ],
       [ 0.        ],
       [ 0.        ]])
```

In [ ]:

```
d_mu=np.dot(np.linalg.inv(H),t)
d_mu
```

Out[ ]:

```
array([[-0.40904134],
       [-0.07923032],
       [ 0.31480962],
       [ 0.17346204],
       [-0.49153583],
       [ 0.04038502]])
```

In [ ]:

```
d_mu.reshape(1,-1)[0][0:4]
```

Out[ ]:

```
array([-0.40904134, -0.07923032,  0.31480962,  0.17346204])
```

In [ ]:

```
print(np.dot(A.T,d_mu[4:]))
print(A.T.shape)
print(d_mu[4:].shape)
```

```
[[-0.45115081]
 [-0.57230588]
 [-0.37038076]
 [-0.65307592]]
(4, 2)
(2, 1)
```

In [ ]:

```
np.dot(A.T,d_mu[4:]).reshape(1,-1)[0]
```

Out[ ]:

```
array([-0.45115081, -0.57230588, -0.37038076, -0.65307592])
```

In [ ]:

```
while np.linalg.norm(gradf(x0)+np.dot(A.T,d_mu[4:]).reshape(1,-1)[0])>0.001 and iter<100:

  x1=x0+d_mu.reshape(1,-1)[0][0:4]
  print("x",iter+1," : ",x1)
  print("f",iter+1," : ",fun(x1))
  print("g",iter+1," : ",gradf(x1))
  print("\n===========\n")

  x0,iter=x1,iter+1
  H1=np.column_stack((Hessian(x0),A.T))
  H2=np.column_stack((A,[0,0],[0,0]))
  H=np.vstack((H1,H2))
  d_mu=np.dot(np.linalg.inv(H),np.vstack((-gradf(x0).reshape(-1,1),[0],[0])))
print("\n===========\n")
print("x",iter," : ",x0)
print("f",iter," : ",fun(x1))
print("g",iter," : ",gradf(x0))
```

```
x 1  :  [0.25762533 0.25410302 0.31480962 0.17346204]
f 1  :  0.7718263578255395
g 1  :  [0.57377007 0.57852657 0.50013952 0.69491102]

===========
x 2  :  [0.27463846 0.22267469 0.31163691 0.19104995]
f 2  :  0.7729264489055946
g 2  :  [0.55116364 0.62215189 0.50405205 0.66826662]

===========


===========

x 2  :  [0.27463846 0.22267469 0.31163691 0.19104995]
f 2  :  0.7729264489055946
g 2  :  [0.55116364 0.62215189 0.50405205 0.66826662]
```

In [ ]:

In [ ]: