IC 152

Computing & Data Science

Lab 8

Mohit Verma

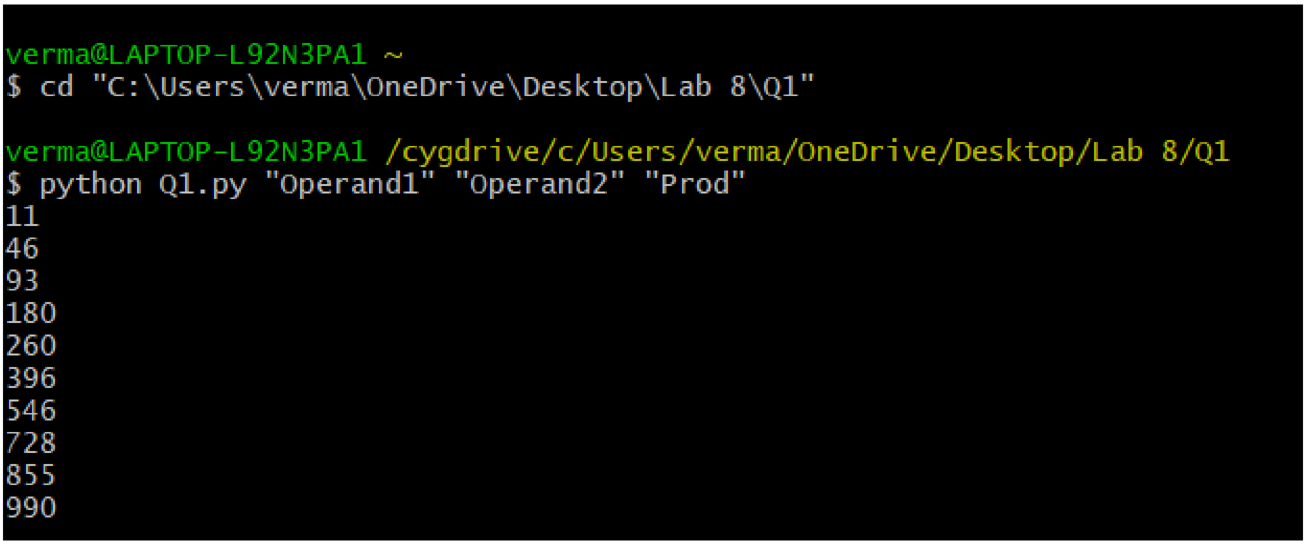
B20215

Instructor- Dr.Padmanabhan Rajan

Questions & Answers

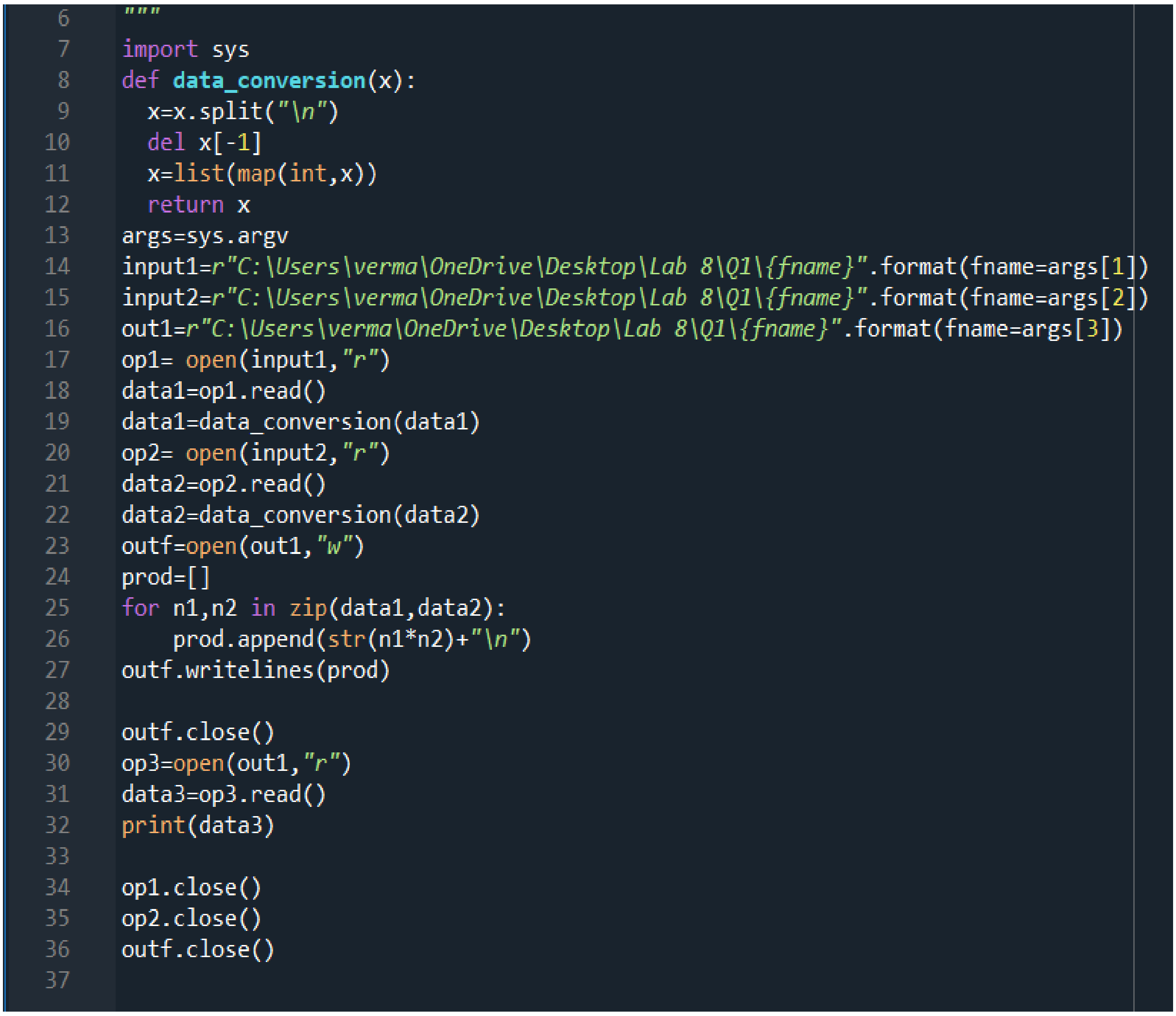
Q1 1. The Files Operand1 and Operand2 contain a list of integers each on a separate line. Read both the files and create a new file Prod. The ith line of the file Prod, should have the product of the elements on the ith lines of file Operand1 and Operand2 respectively.

Ans 1) The output is shown below-



```
verma@LAPTOP-L92N3PA1 ~
$ cd "C:\Users\verma\OneDrive\Desktop\Lab 8\Q1"

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q1
$ python Q1.py "Operand1" "Operand2" "Prod"
11
46
93
180
260
396
546
728
855
990
```

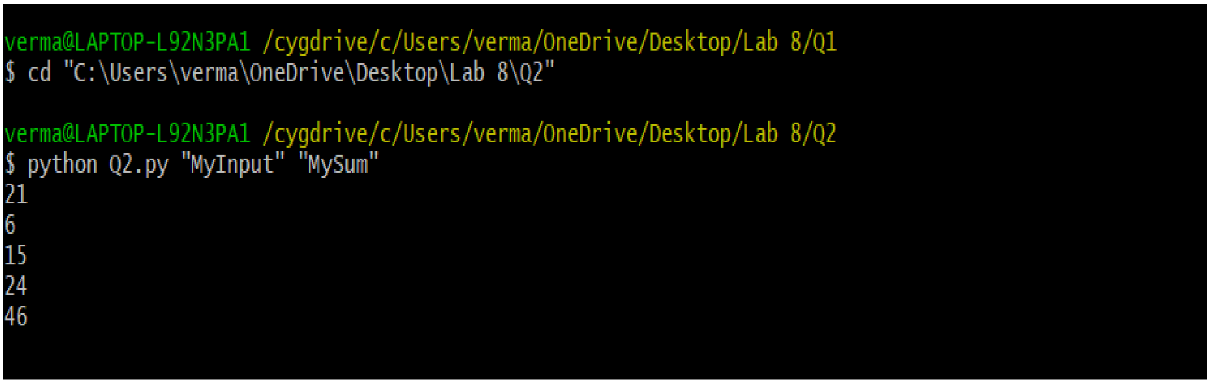
The code for output is shown below-



```python
"""
import sys
def data_conversion(x):
  x=x.split("\n")
  del x[-1]
  x=list(map(int,x))
  return x
args=sys.argv
input1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q1\{fname}".format(fname=args[1])
input2=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q1\{fname}".format(fname=args[2])
out1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q1\{fname}".format(fname=args[3])
op1= open(input1,"r")
data1=op1.read()
data1=data_conversion(data1)
op2= open(input2,"r")
data2=op2.read()
data2=data_conversion(data2)
outf=open(out1,"w")
prod=[]
for n1,n2 in zip(data1,data2):
    prod.append(str(n1*n2)+"\n")
outf.writelines(prod)

outf.close()
op3=open(out1,"r")
data3=op3.read()
print(data3)

op1.close()
op2.close()
outf.close()

```

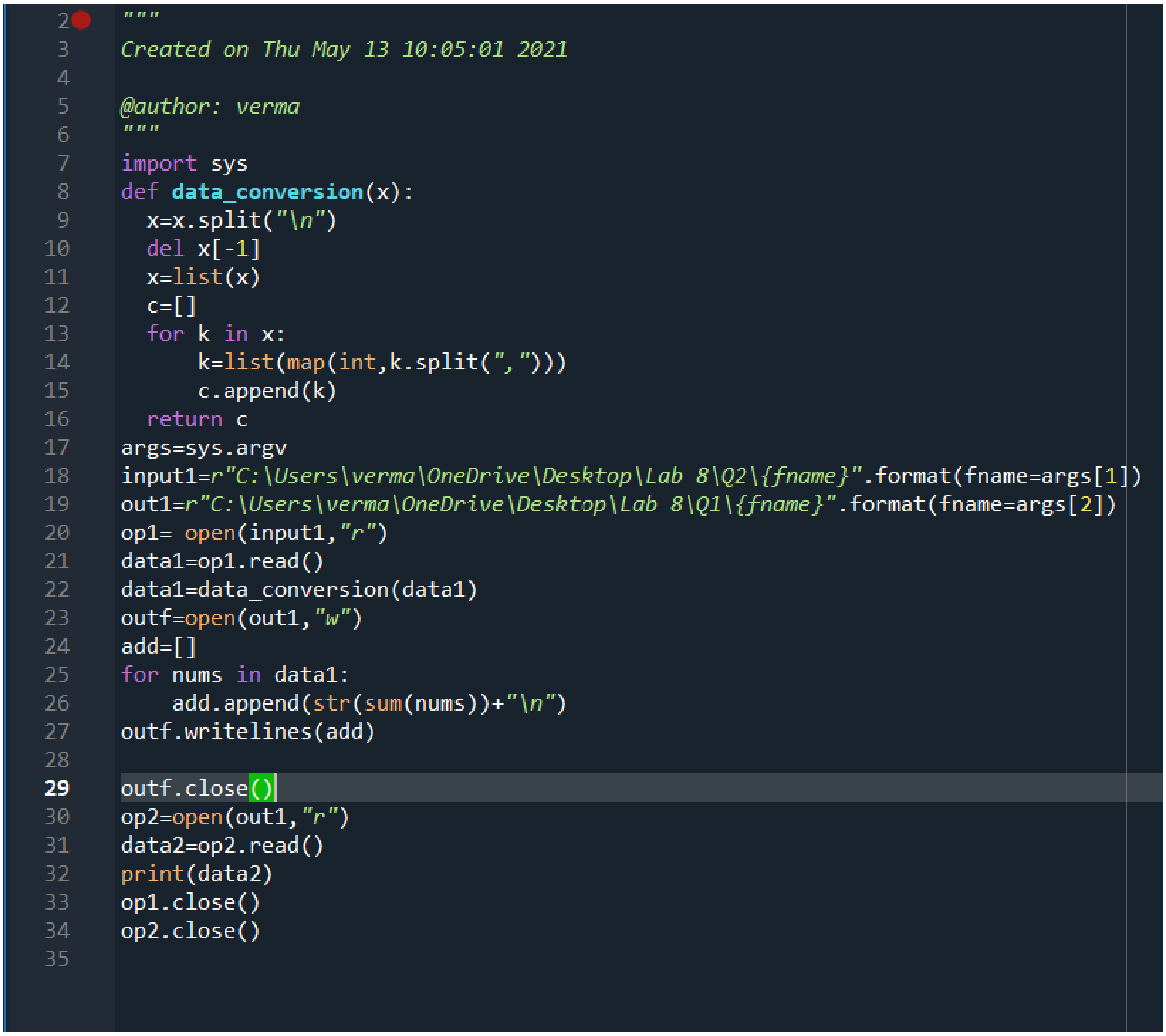
Q2) MyInput contains several lines. Each line has arbitrary number of integers separated by a comma. The goal is to create a file MySum such that the ith line of MySum contains the sum of all the integers on the ith line of MyInput.

Ans 2) The output is shown below-



```
verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q1
$ cd "C:\Users\verma\OneDrive\Desktop\Lab 8\Q2"

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q2
$ python Q2.py "MyInput" "MySum"
21
6
15
24
46
```

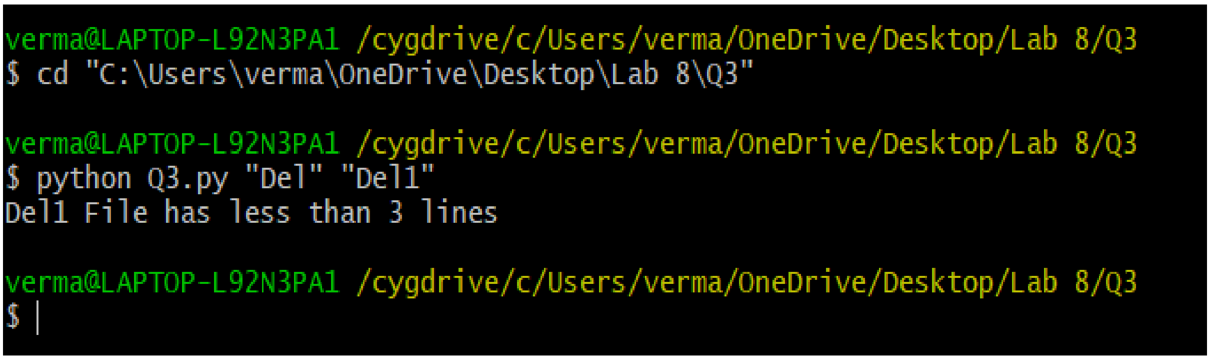
The code for output is shown below-



```
"""
Created on Thu May 13 10:05:01 2021

@author: verma
"""
import sys
def data_conversion(x):
 x=x.split("\n")
 del x[-1]
 x=list(x)
 c=[]
 for k in x:
   k=list(map(int,k.split(",")))
   c.append(k)
 return c
args=sys.argv
input1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q2\{fname}".format(fname=args[1])
out1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q1\{fname}".format(fname=args[2])
op1= open(input1,"r")
data1=op1.read()
data1=data_conversion(data1)
outf=open(out1,"w")
add=[]
for nums in data1:
  add.append(str(sum(nums))+"\n")
outf.writelines(add)

outf.close()
op2=open(out1,"r")
data2=op2.read()
print(data2)
op1.close()
op2.close()

```

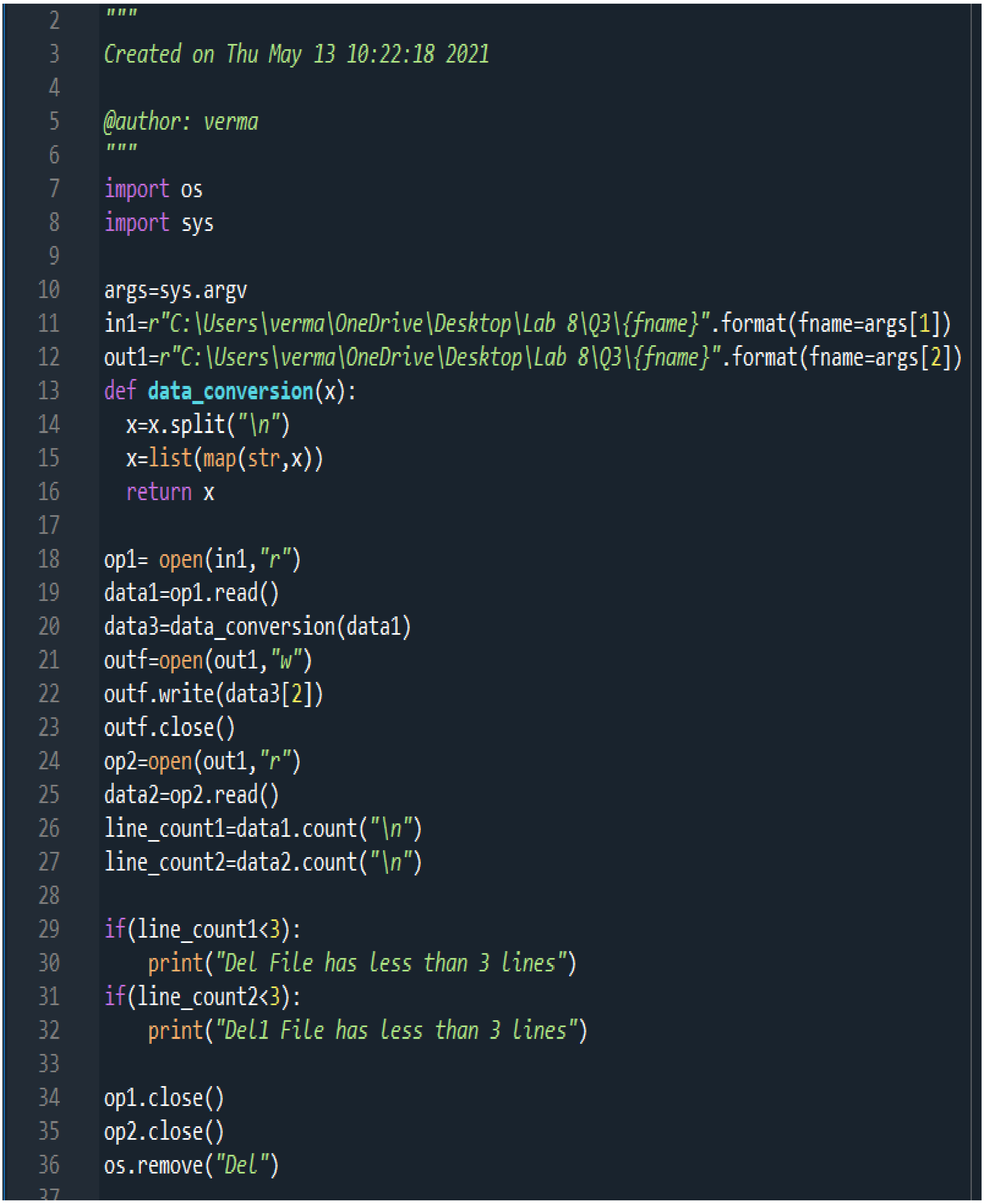
Q3) Del is file stored in the local folder. Create a new file Del1 which contains only the contents of the 3rd line of the file Del and then delete the file Del by importing OS module. If the Del file has less than three lines, then it should output "File has less than 3 lines" on the terminal (without quotes) and the same to Del1 file.

Ans 3) The output is shown below--



```
verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q3
$ cd "C:\Users\verma\OneDrive\Desktop\Lab 8\Q3"

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q3
$ python Q3.py "Del" "Del1"
Del1 File has less than 3 lines

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q3
$ |
```

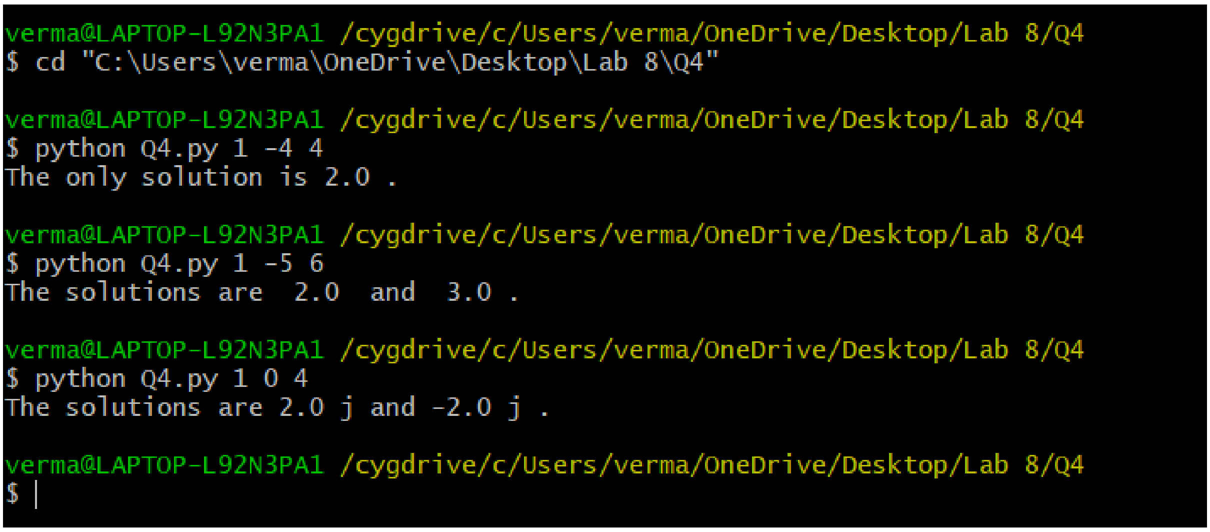
The code for output is shown below-



```
"""
Created on Thu May 13 10:22:18 2021

@author: verma
"""
import os
import sys

args=sys.argv
in1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q3\{fname}".format(fname=args[1])
out1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q3\{fname}".format(fname=args[2])
def data_conversion(x):
  x=x.split("\n")
  x=list(map(str,x))
  return x

op1= open(in1,"r")
data1=op1.read()
data3=data_conversion(data1)
outf=open(out1,"w")
outf.write(data3[2])
outf.close()
op2=open(out1,"r")
data2=op2.read()
line_count1=data1.count("\n")
line_count2=data2.count("\n")

if(line_count1<3):
    print("Del File has less than 3 lines")
if(line_count2<3):
    print("Del1 File has less than 3 lines")

op1.close()
op2.close()
os.remove("Del")
```

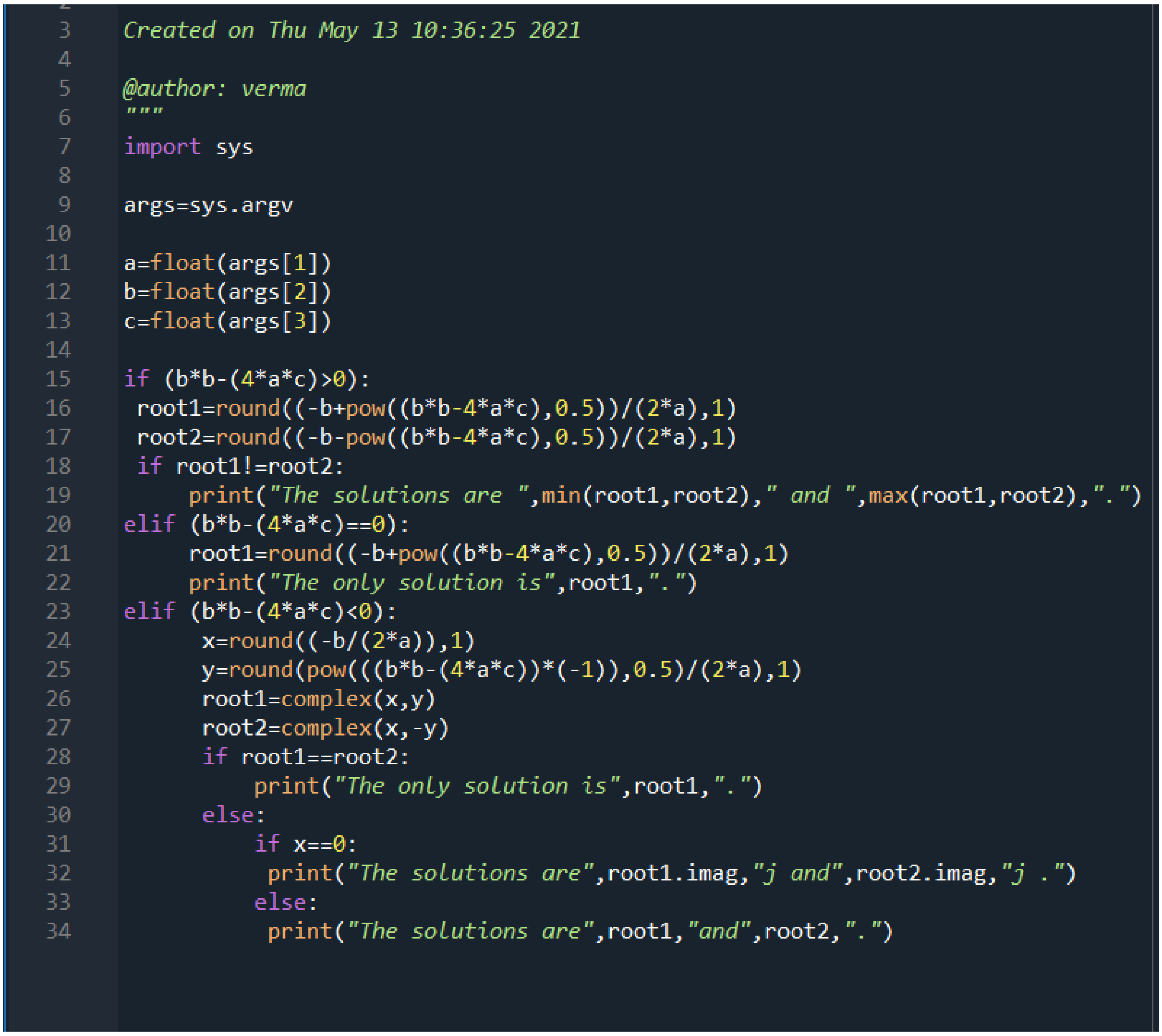
Q4) Let quad.py solve quadratic equations of the form ax2 + bx + c = 0, use a command line argument to pass the arguments a; b and c to get the solution for the corresponding quadratic equation.

Ans 4) The output for the problem is shown below-



```
verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q4
$ cd "C:\Users\verma\OneDrive\Desktop\Lab 8\Q4"

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q4
$ python Q4.py 1 -4 4
The only solution is 2.0 .

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q4
$ python Q4.py 1 -5 6
The solutions are  2.0  and  3.0 .

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q4
$ python Q4.py 1 0 4
The solutions are 2.0 j and -2.0 j .

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q4
$ |
```

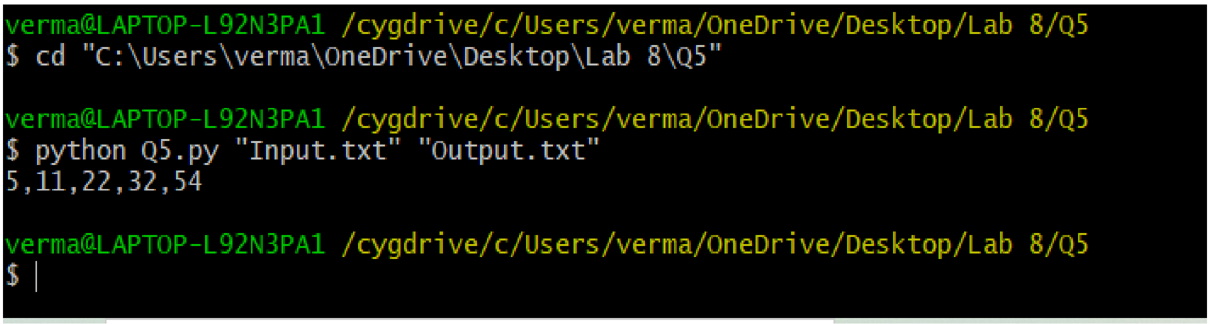
The code for output is shown below-



```python
Created on Thu May 13 10:36:25 2021

@author: verma
"""
import sys

args=sys.argv

a=float(args[1])
b=float(args[2])
c=float(args[3])

if (b*b-(4*a*c)>0):
 root1=round((-b+pow((b*b-4*a*c),0.5))/(2*a),1)
 root2=round((-b-pow((b*b-4*a*c),0.5))/(2*a),1)
 if root1!=root2:
     print("The solutions are ",min(root1,root2)," and ",max(root1,root2),".")
elif (b*b-(4*a*c)==0):
     root1=round((-b+pow((b*b-4*a*c),0.5))/(2*a),1)
     print("The only solution is",root1,".")
elif (b*b-(4*a*c)<0):
      x=round((-b/(2*a)),1)
      y=round(pow(((b*b-(4*a*c))*(-1)),0.5)/(2*a),1)
      root1=complex(x,y)
      root2=complex(x,-y)
      if root1==root2:
          print("The only solution is",root1,".")
      else:
          if x==0:
           print("The solutions are",root1.imag,"j and",root2.imag,"j .")
          else:
           print("The solutions are",root1,"and",root2,".")
```

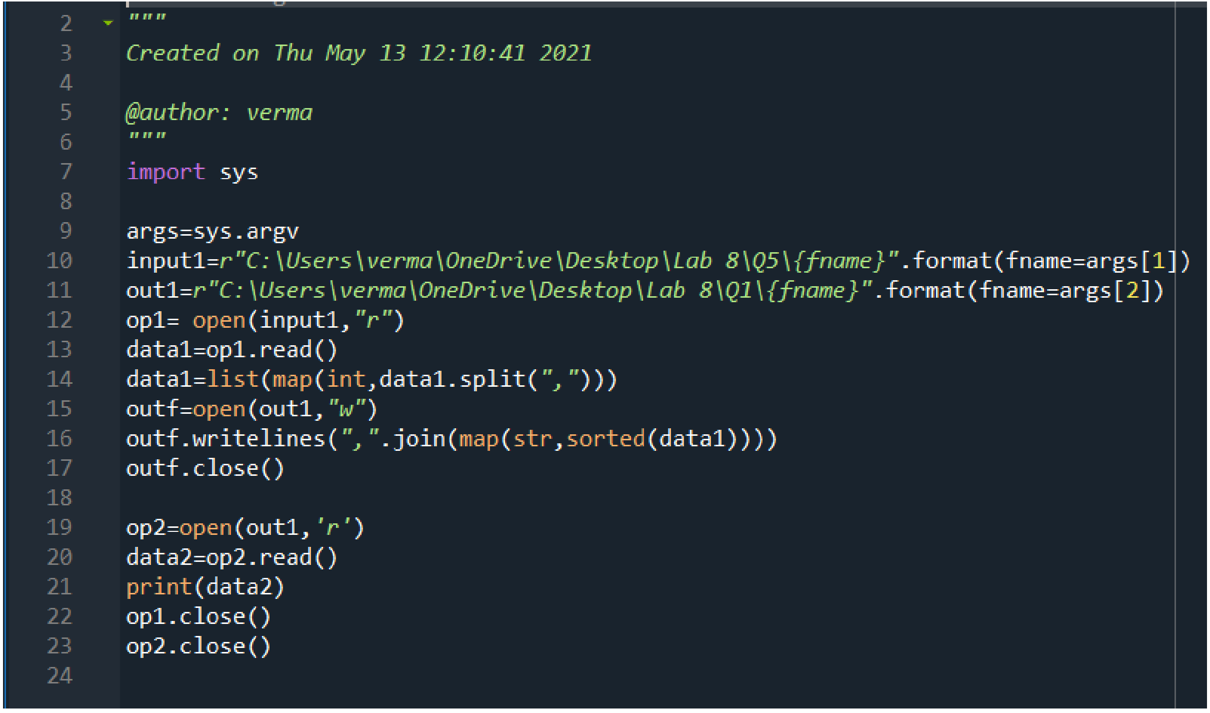
Q5) Sort.py contains some code to sort n integers. You can use library function “. sort ()”. A file named Input contains a list of integers separated by a comma. Use a command line argument to Sort elements in input file using Sort.py and output the sorted order.

Ans 5) The output is shown below-



```
verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q5
$ cd "C:\Users\verma\OneDrive\Desktop\Lab 8\Q5"

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q5
$ python Q5.py "Input.txt" "Output.txt"
5,11,22,32,54

verma@LAPTOP-L92N3PA1 /cygdrive/c/Users/verma/OneDrive/Desktop/Lab 8/Q5
$ |
```

The code for output is shown below-



```
"""
Created on Thu May 13 12:10:41 2021

@author: verma
"""
import sys

args=sys.argv
input1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q5\{fname}".format(fname=args[1])
out1=r"C:\Users\verma\OneDrive\Desktop\Lab 8\Q1\{fname}".format(fname=args[2])
op1= open(input1,"r")
data1=op1.read()
data1=list(map(int,data1.split(",")))
outf=open(out1,"w")
outf.writelines(",".join(map(str,sorted(data1))))
outf.close()

op2=open(out1,'r')
data2=op2.read()
print(data2)
op1.close()
op2.close()
```

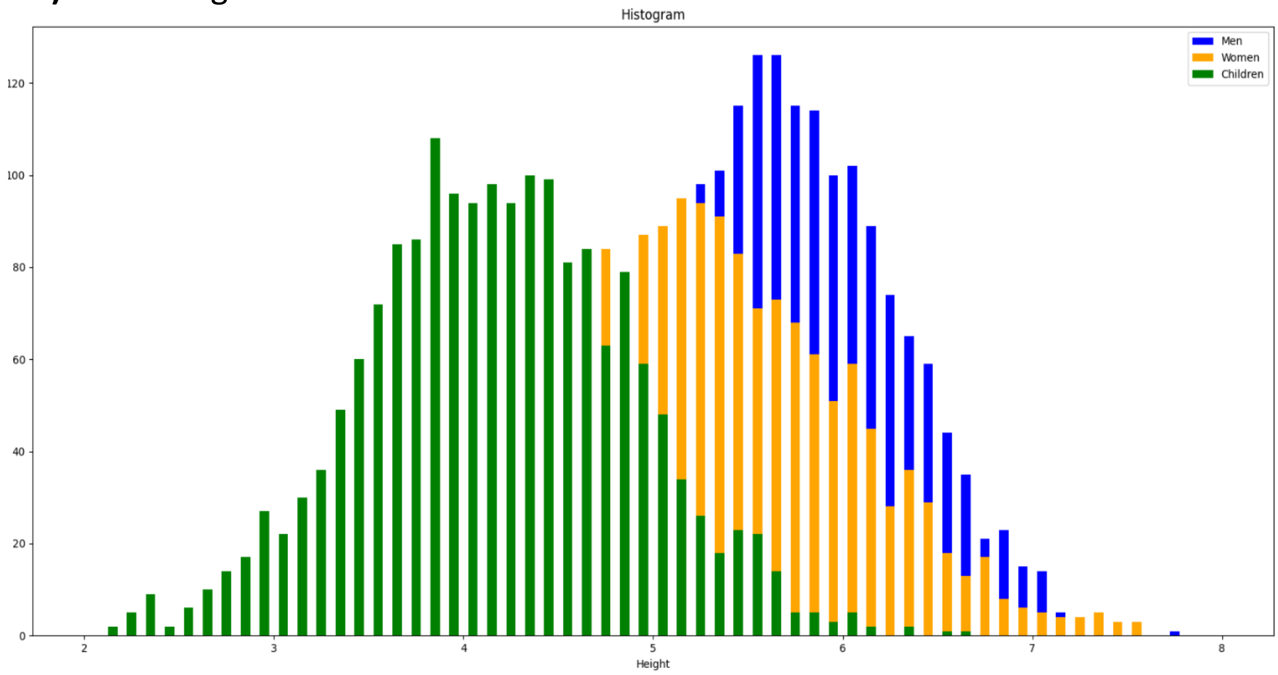
Q6) A worldwide health club chain has men, women and children as members. A survey of the heights of all members was done, and the histogram is plotted. The histogram has three modes (or peaks.)

a) Explain why there are three modes in the histogram.

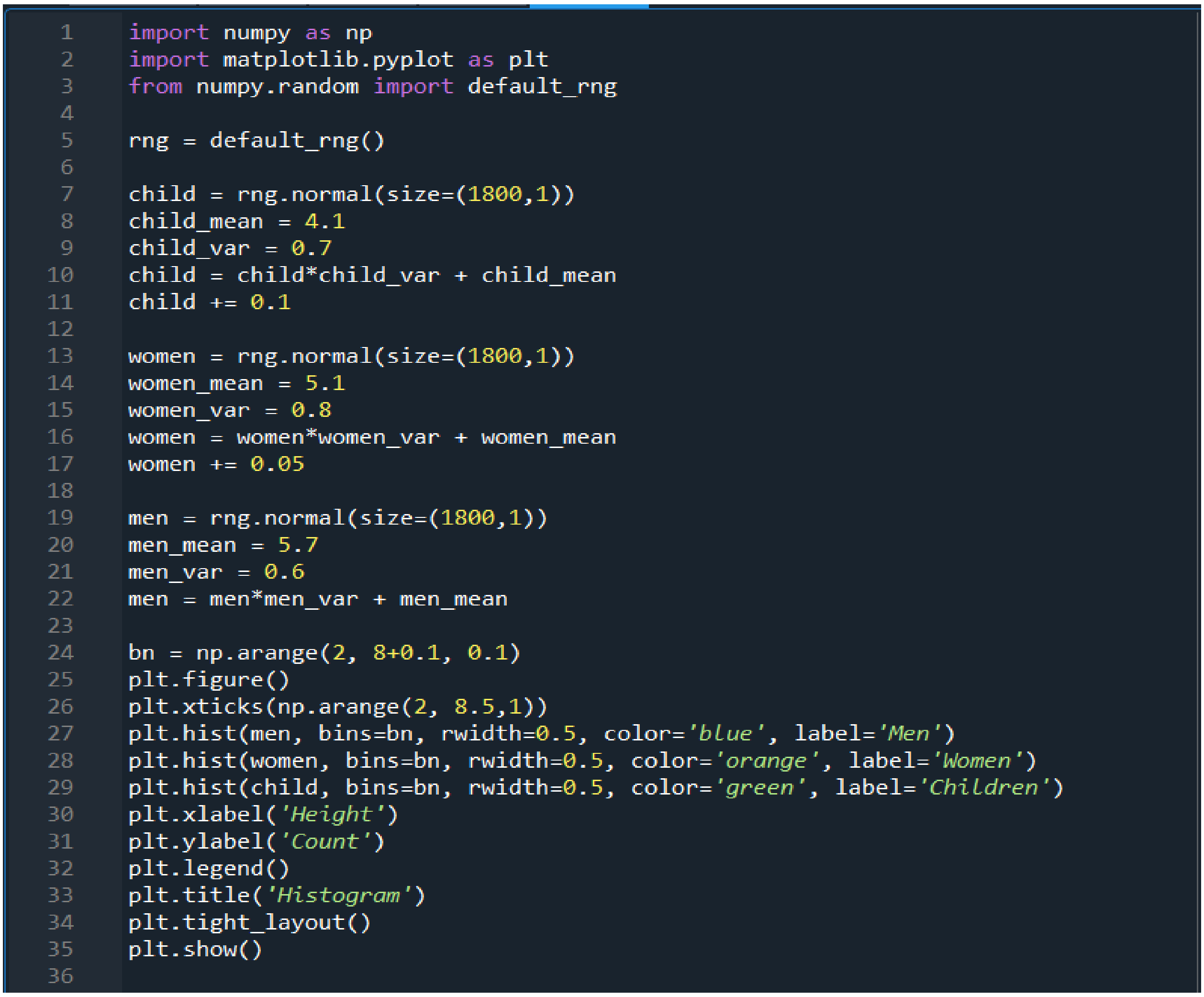
b) Write a Python program that will simulate the data represented by this histogram. Assume that the variation in height is largest for men and the smallest for children.

Ans) a) There are three modes in the data since we have three different types of data of Men, Women, Child plotted on same histogram. Hence, we have a multimodal data which will have modes according to the constituent data types.

b) The histogram is shown below-



The code for output is shown below-



```python
import numpy as np
import matplotlib.pyplot as plt
from numpy.random import default_rng

rng = default_rng()

child = rng.normal(size=(1800,1))
child_mean = 4.1
child_var = 0.7
child = child*child_var + child_mean
child += 0.1

women = rng.normal(size=(1800,1))
women_mean = 5.1
women_var = 0.8
women = women*women_var + women_mean
women += 0.05

men = rng.normal(size=(1800,1))
men_mean = 5.7
men_var = 0.6
men = men*men_var + men_mean

bn = np.arange(2, 8+0.1, 0.1)
plt.figure()
plt.xticks(np.arange(2, 8.5,1))
plt.hist(men, bins=bn, rwidth=0.5, color='blue', label='Men')
plt.hist(women, bins=bn, rwidth=0.5, color='orange', label='Women')
plt.hist(child, bins=bn, rwidth=0.5, color='green', label='Children')
plt.xlabel('Height')
plt.ylabel('Count')
plt.legend()
plt.title('Histogram')
plt.tight_layout()
plt.show()
```

Q7) Accept the following input from the user:

- **Num rows, M**
- **Num columns, N**
- **Range, R**

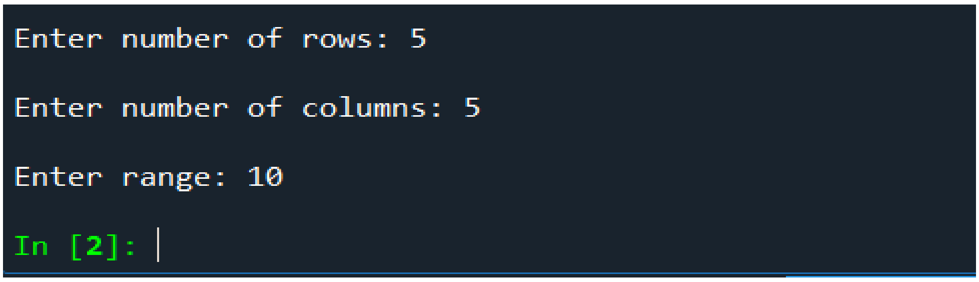
(a) Generate a random MxN matrix consisting of uniformly distributed random integers in the range 0 and R (both inclusive) and determine its Frobenius norm. Repeat this experiment 1000 times.

(b) Plot the histogram of the norms.

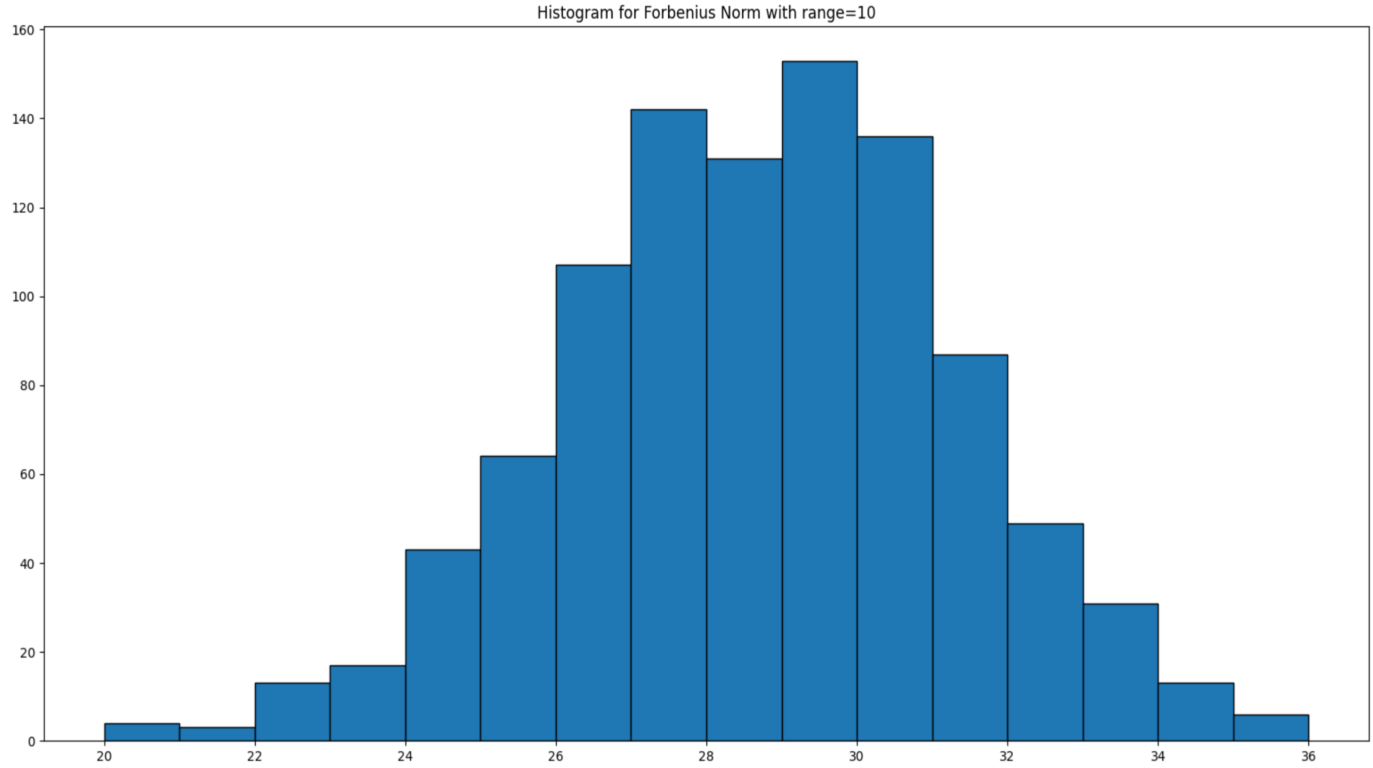
(c) Repeat the above-mentioned experiment for the following values of R in the set {5, 10, 50, 500, 1000} and plot box plots of the norm for the various values of R.

(d) Does the result change if the matrix entries are normally distributed?

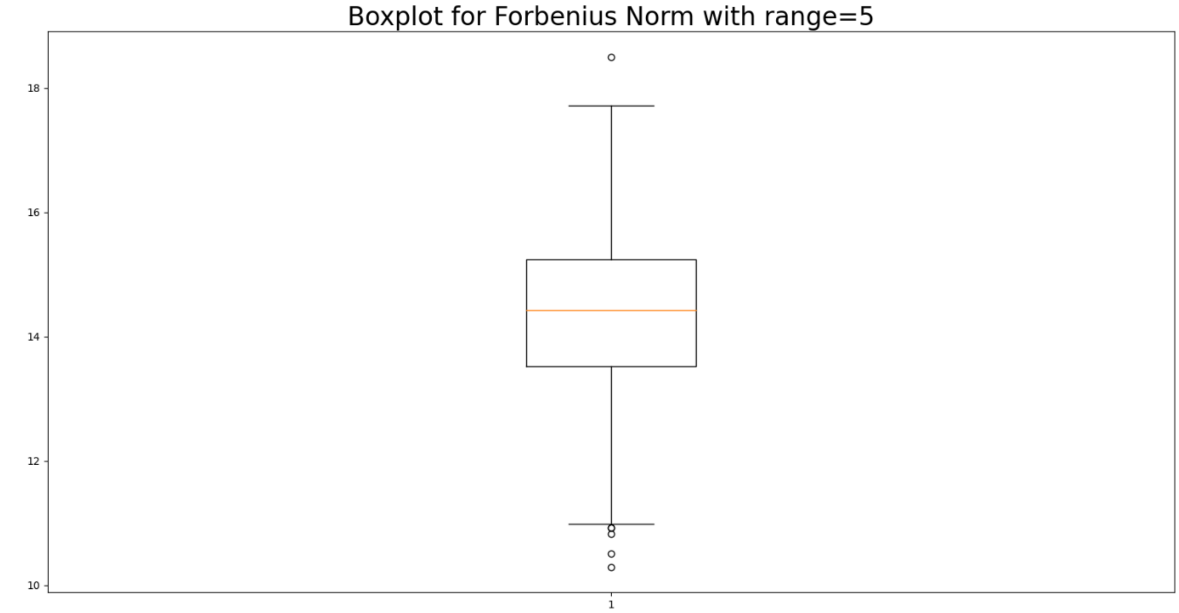
Ans a & b) The matrix is generated and corresponding histogram for **part (b)** is shown below-

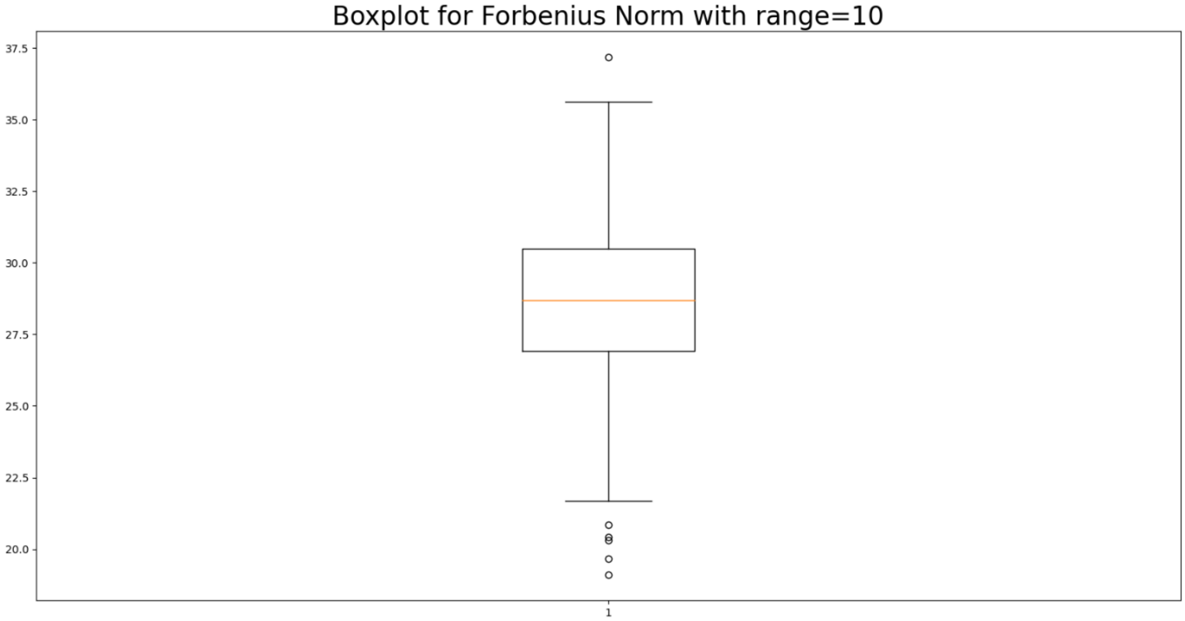


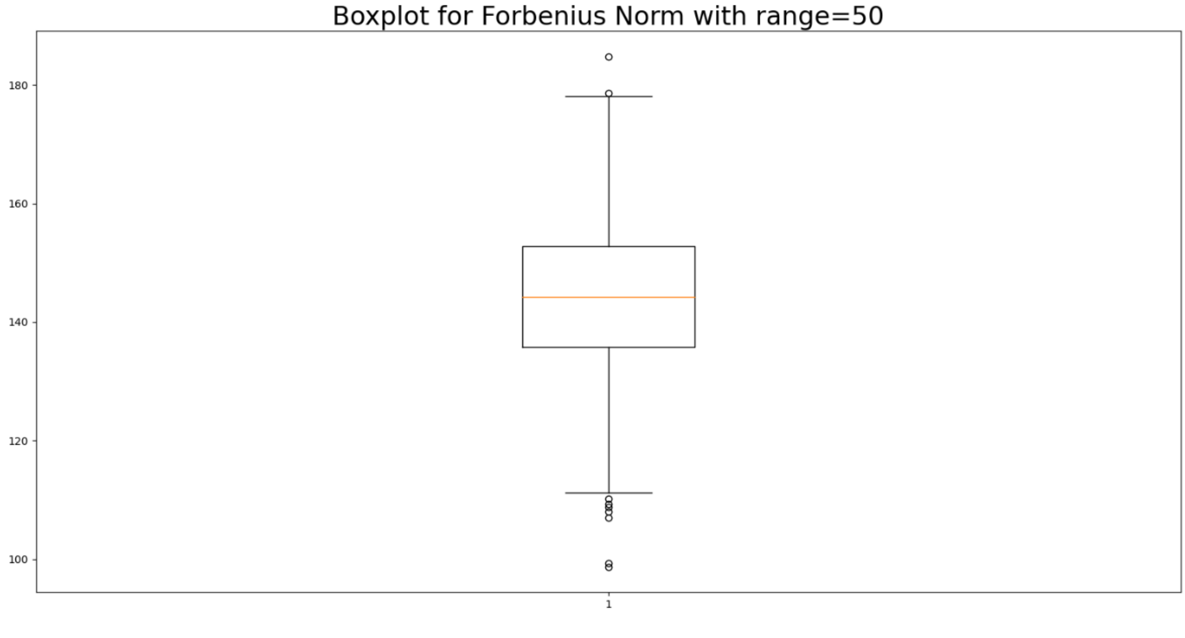
```
Enter number of rows: 5

Enter number of columns: 5

Enter range: 10

In [2]:
```

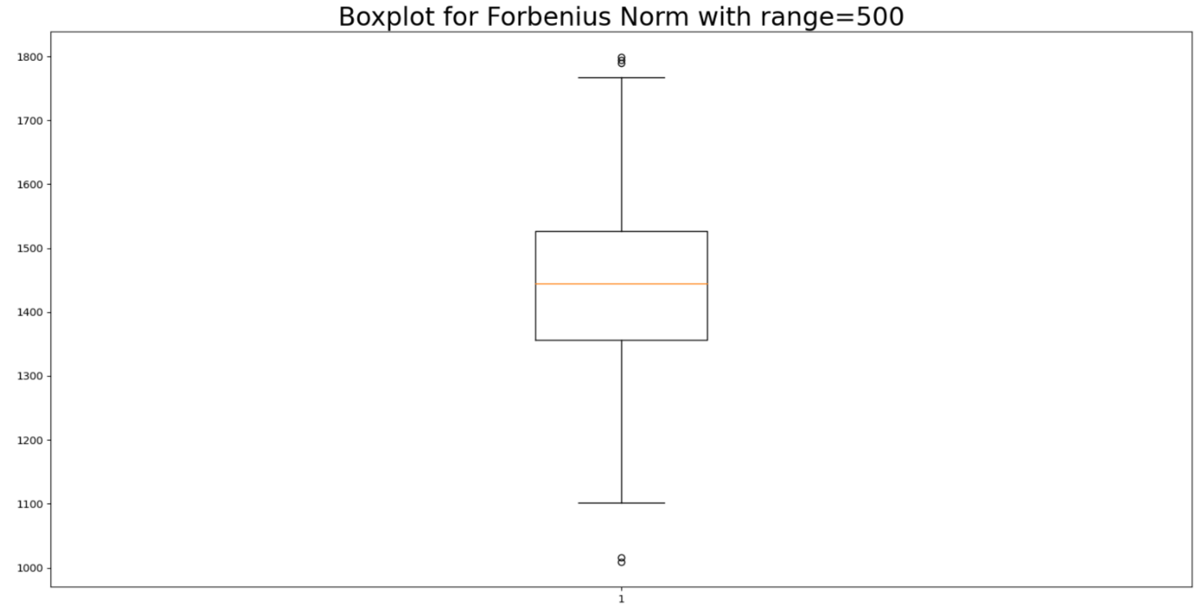


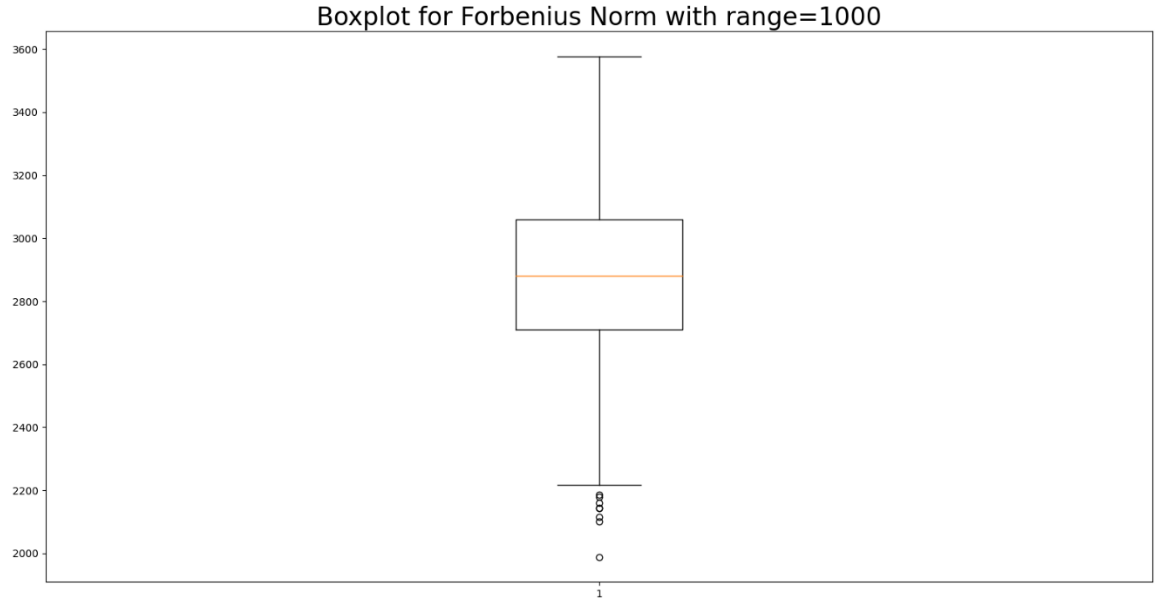
(c) The box plots are shown below –


Boxplot for Forbenius Norm with range=5

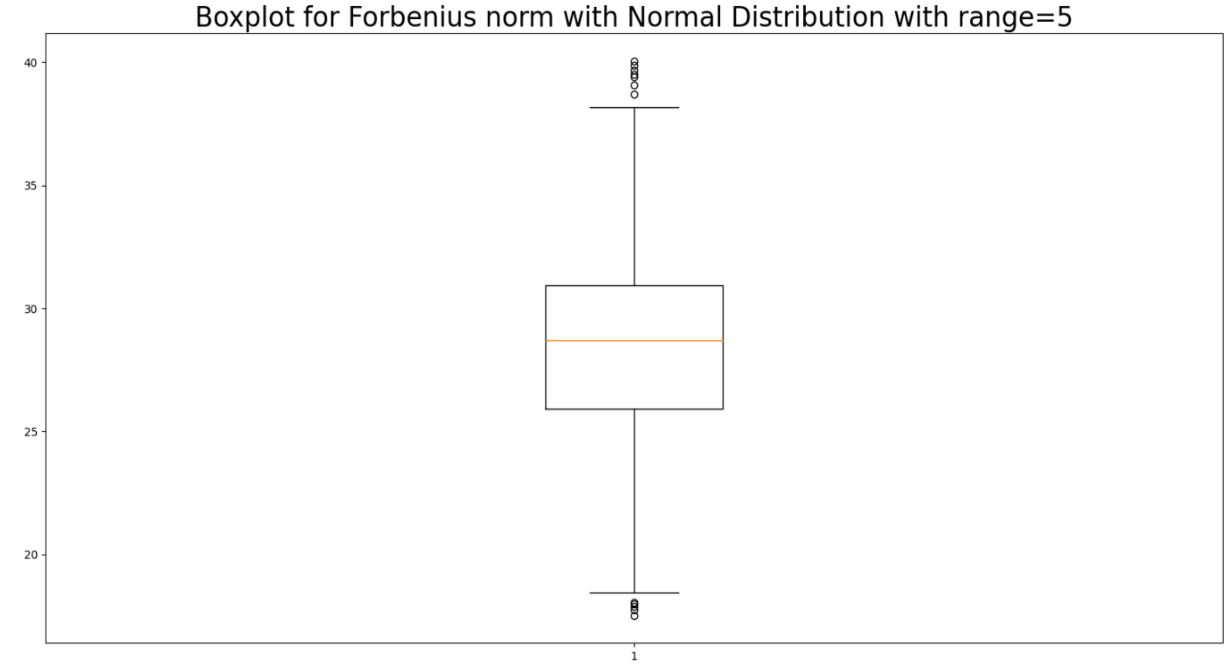

Boxplot for Forbenius Norm with range=10


Boxplot for Forbenius Norm with range=50

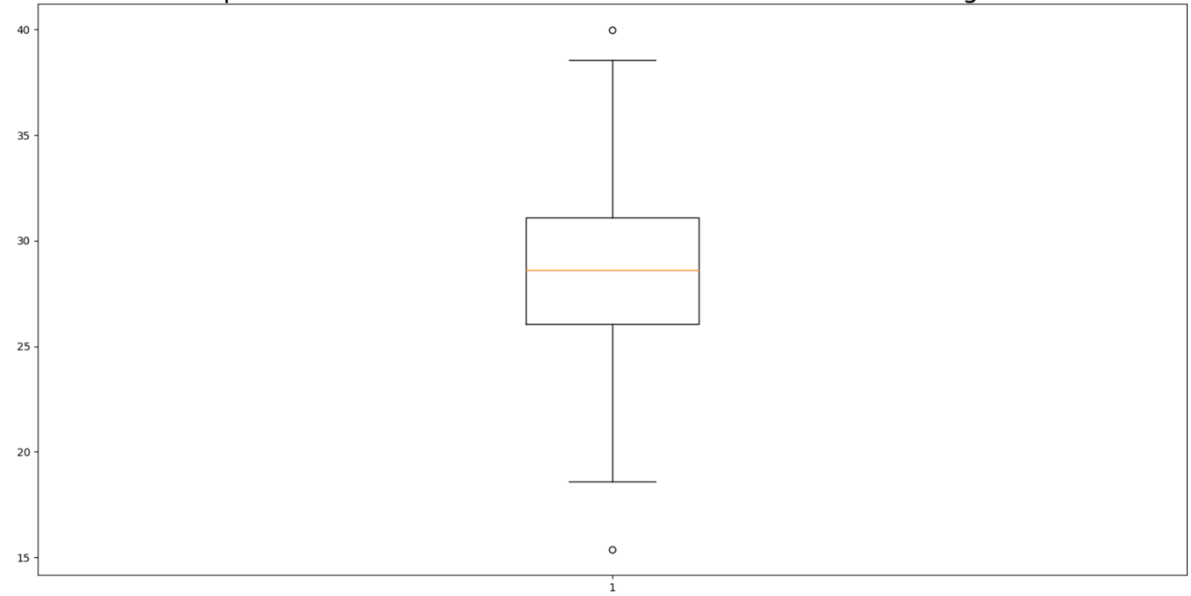




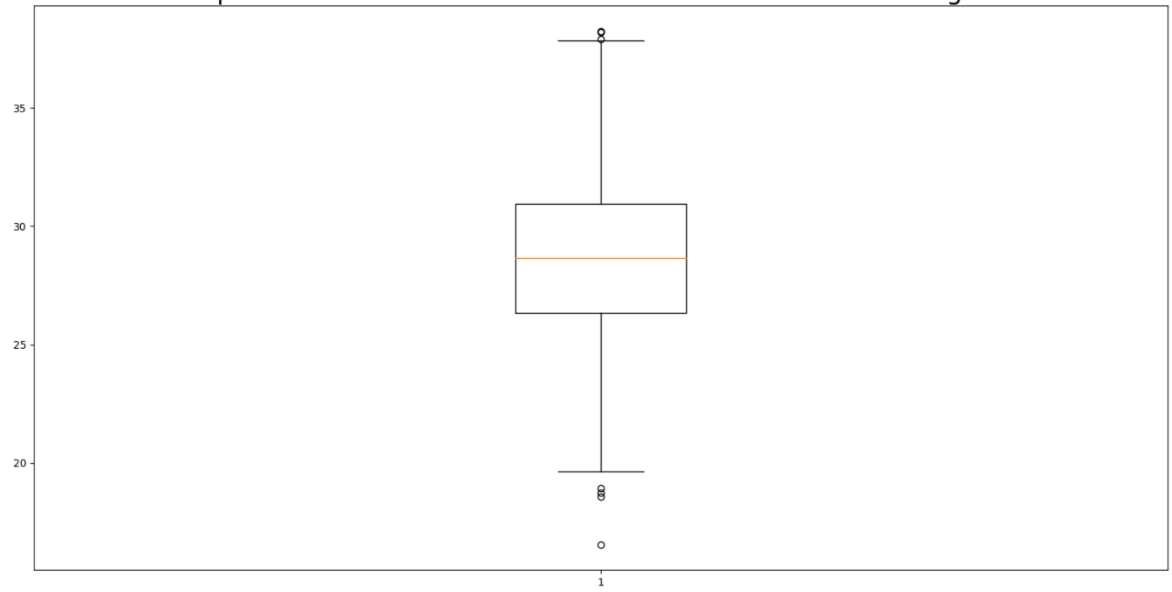
(d) Yes, the results changes if the entries of the matrix is normally distributed. The following boxplots show it-



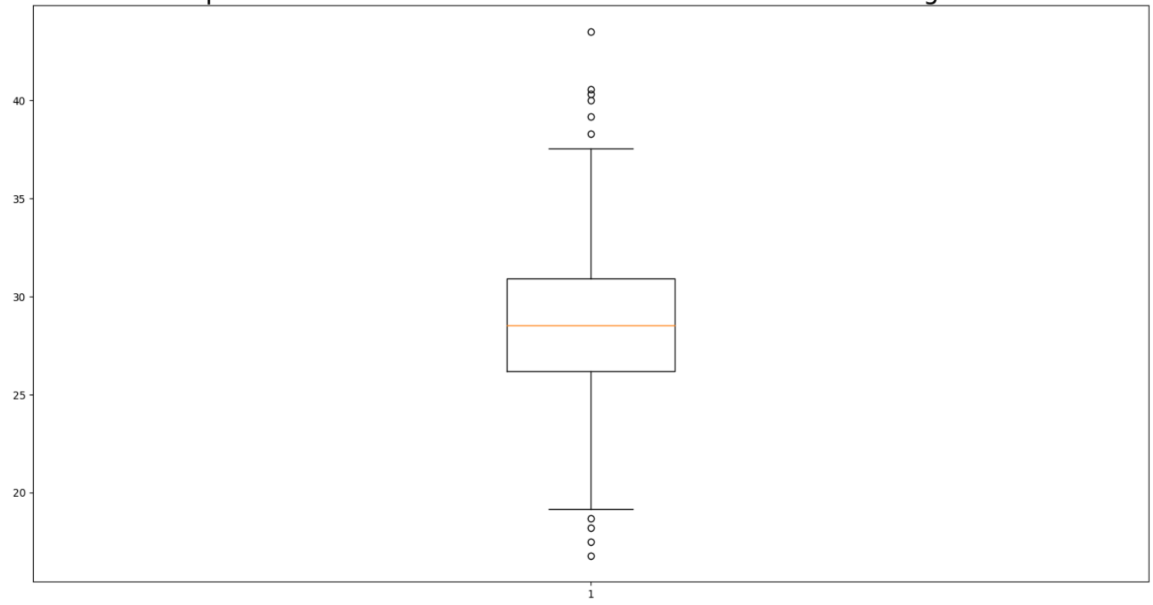
Boxplot for Forbenius norm with Normal Distribution with range=10

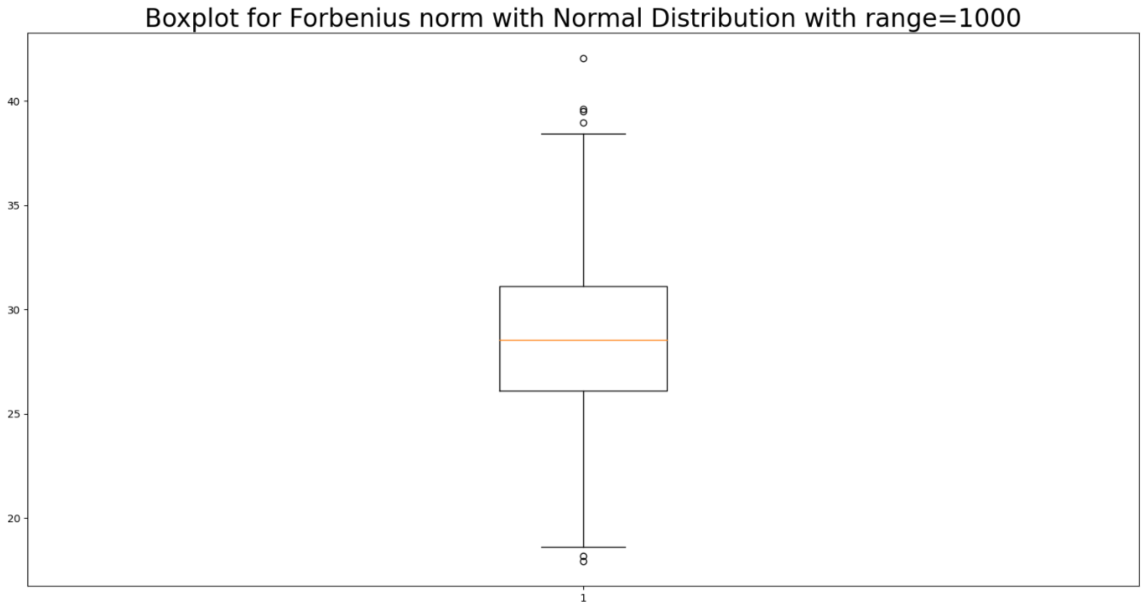


Boxplot for Forbenius norm with Normal Distribution with range=50



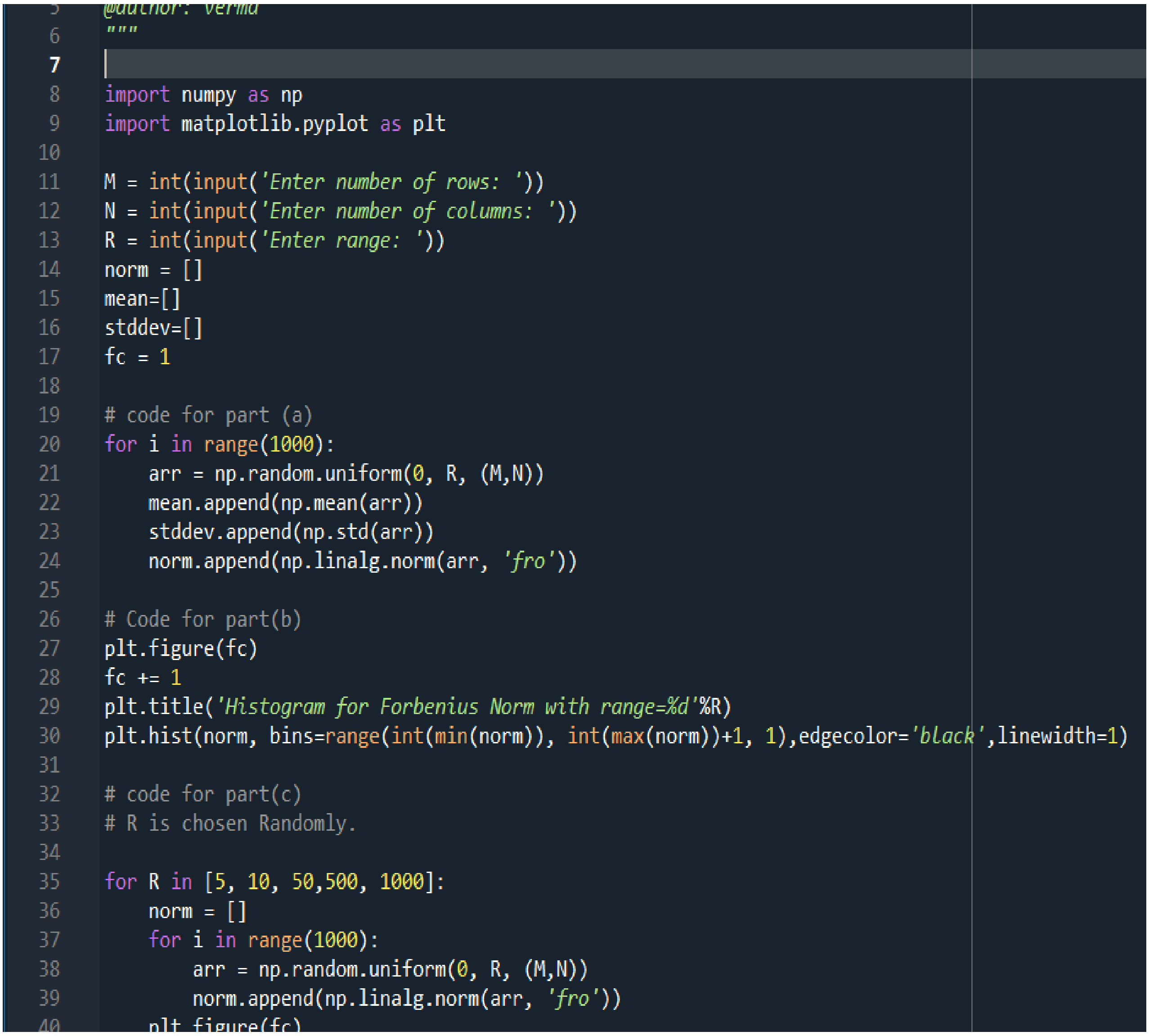
Boxplot for Forbenius norm with Normal Distribution with range=500



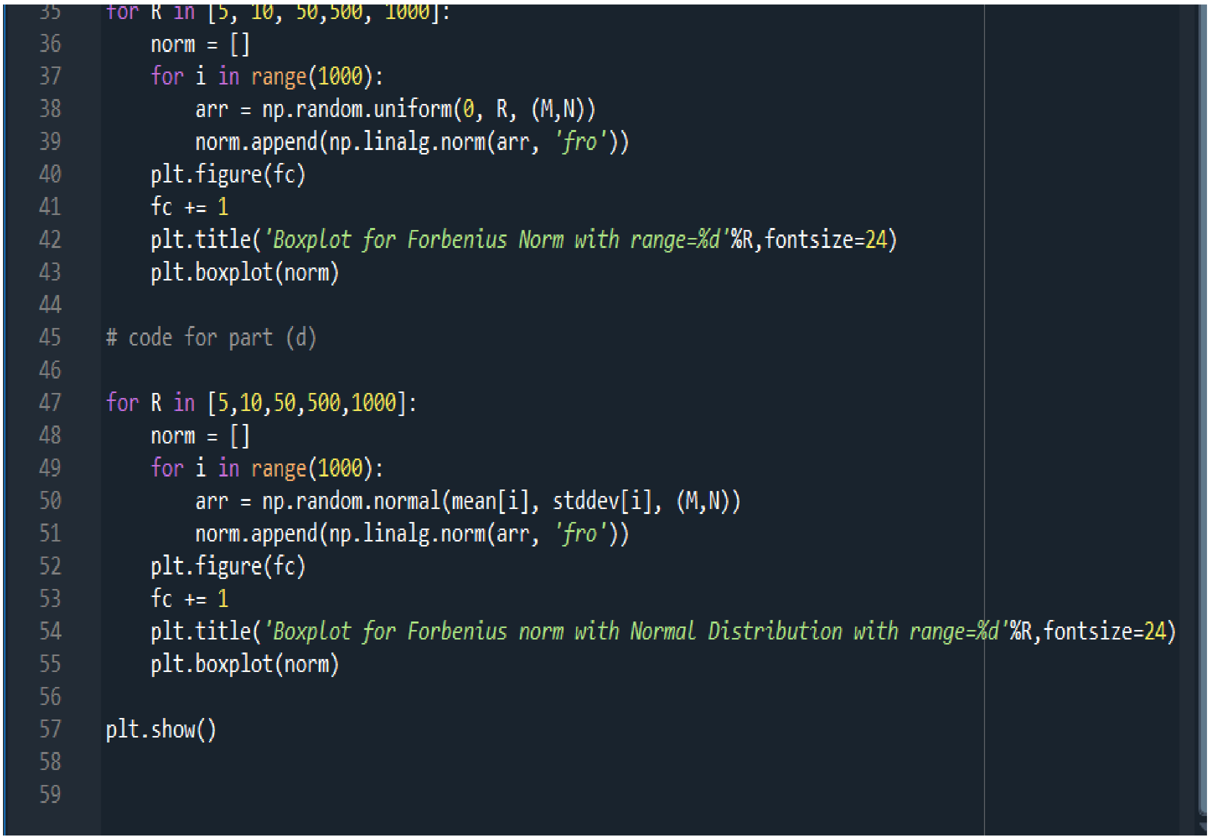


Boxplot for Forbenius norm with Normal Distribution with range=1000

The code for all part is shown below-



```
@author: Verma
"""

import numpy as np
import matplotlib.pyplot as plt

M = int(input('Enter number of rows: '))
N = int(input('Enter number of columns: '))
R = int(input('Enter range: '))
norm = []
mean=[]
stddev=[]
fc = 1

# code for part (a)
for i in range(1000):
    arr = np.random.uniform(0, R, (M,N))
    mean.append(np.mean(arr))
    stddev.append(np.std(arr))
    norm.append(np.linalg.norm(arr, 'fro'))

# Code for part(b)
plt.figure(fc)
fc += 1
plt.title('Histogram for Forbenius Norm with range=%d'%R)
plt.hist(norm, bins=range(int(min(norm)), int(max(norm))+1, 1),edgecolor='black',linewidth=1)

# code for part(c)
# R is chosen Randomly.

for R in [5, 10, 50,500, 1000]:
    norm = []
    for i in range(1000):
        arr = np.random.uniform(0, R, (M,N))
        norm.append(np.linalg.norm(arr, 'fro'))
    plt.figure(fc)
```



```
for R in [5, 10, 50,500, 1000]:
    norm = []
    for i in range(1000):
        arr = np.random.uniform(0, R, (M,N))
        norm.append(np.linalg.norm(arr, 'fro'))
    plt.figure(fc)
    fc += 1
    plt.title('Boxplot for Forbenius Norm with range=%d'%R,fontsize=24)
    plt.boxplot(norm)

# code for part (d)

for R in [5,10,50,500,1000]:
    norm = []
    for i in range(1000):
        arr = np.random.normal(mean[i], stddev[i], (M,N))
        norm.append(np.linalg.norm(arr, 'fro'))
    plt.figure(fc)
    fc += 1
    plt.title('Boxplot for Forbenius norm with Normal Distribution with range=%d'%R,fontsize=24)
    plt.boxplot(norm)

plt.show()
```

...