PART
B

(PART B : TO BE COMPLETED BY STUDENTS)

Roll No. C035	Name: Krisha Goti
Class : B	Batch : B1
Date of Experiment: 13/08/2022	Date of Submission:15/08/2022
Grade :	

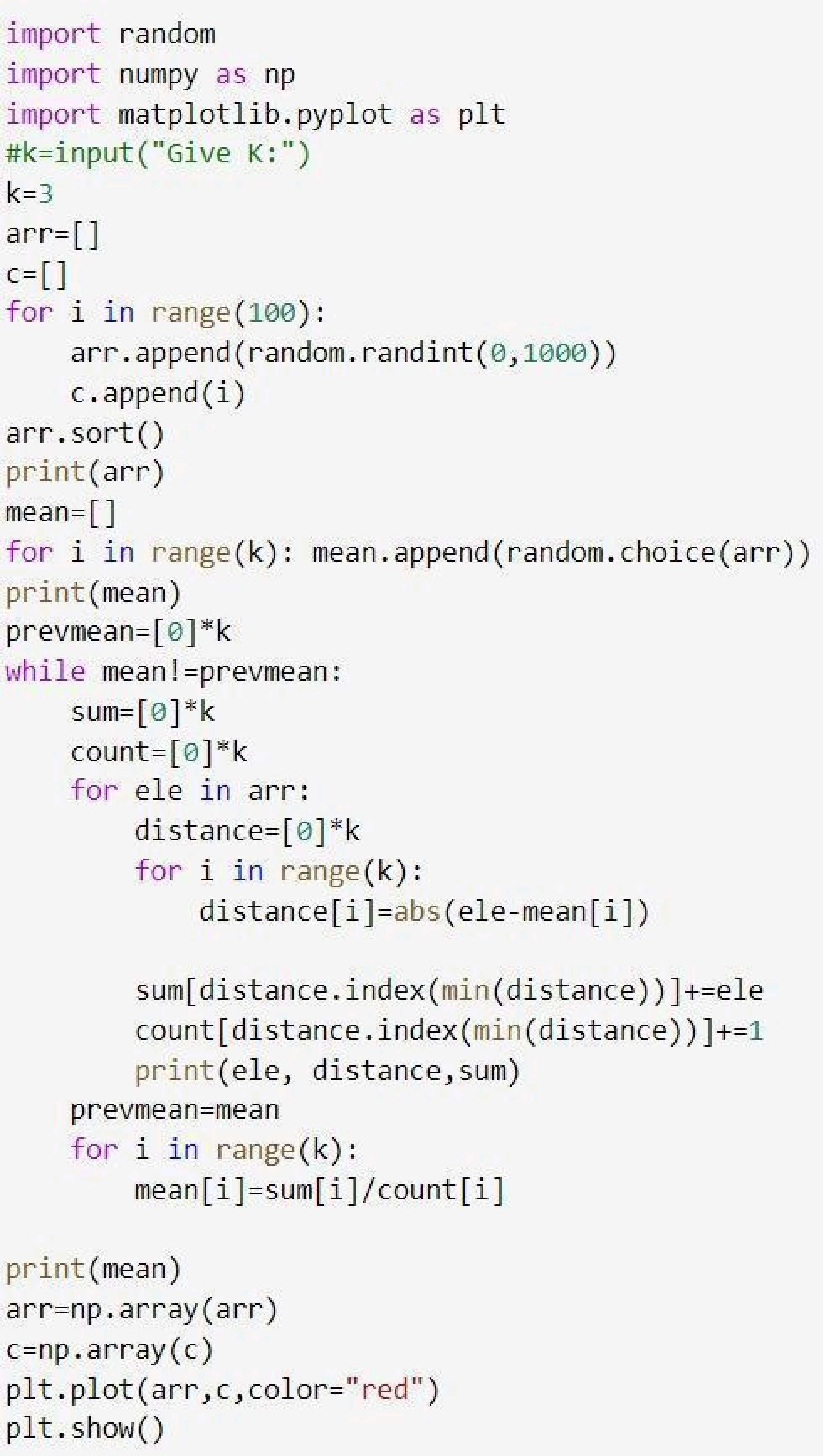
B.1 Clustering Code written by student:

```
import random
import numpy as np
import matplotlib.pyplot as plt
#k=input("Give K:")
k=3
arr=[]
c=[]
for i in range(100):
    arr.append(random.randint(0,1000))
    c.append(i)
arr.sort()
print(arr)
mean=[]
for i in range(k): mean.append(random.choice(arr))
print(mean)
prevmean=[0]*k
while mean!=prevmean:
    sum=[0]*k
    count=[0]*k
    for ele in arr:
        distance=[0]*k
        for i in range(k):
            distance[i]=abs(ele-mean[i])

        sum[distance.index(min(distance))]+=ele
        count[distance.index(min(distance))]+=1
        print(ele, distance,sum)
    prevmean=mean
    for i in range(k):
        mean[i]=sum[i]/count[i]

print(mean)
arr=np.array(arr)
c=np.array(c)
```

```
plt.plot(arr,c,color="red")
plt.show()
```

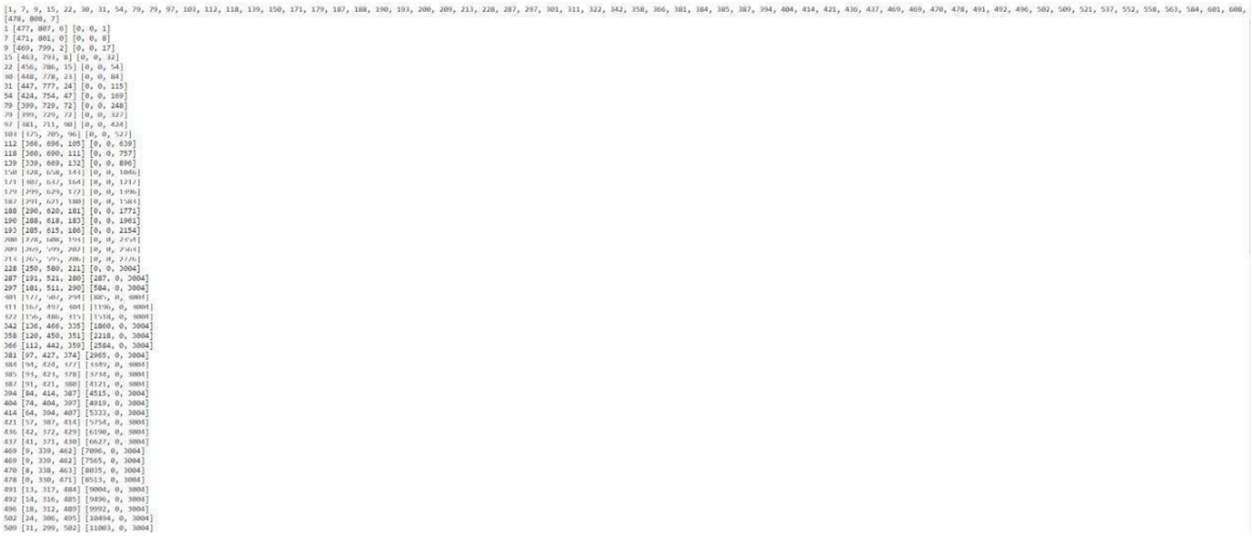
B.2 Input and Output:

Input Data:

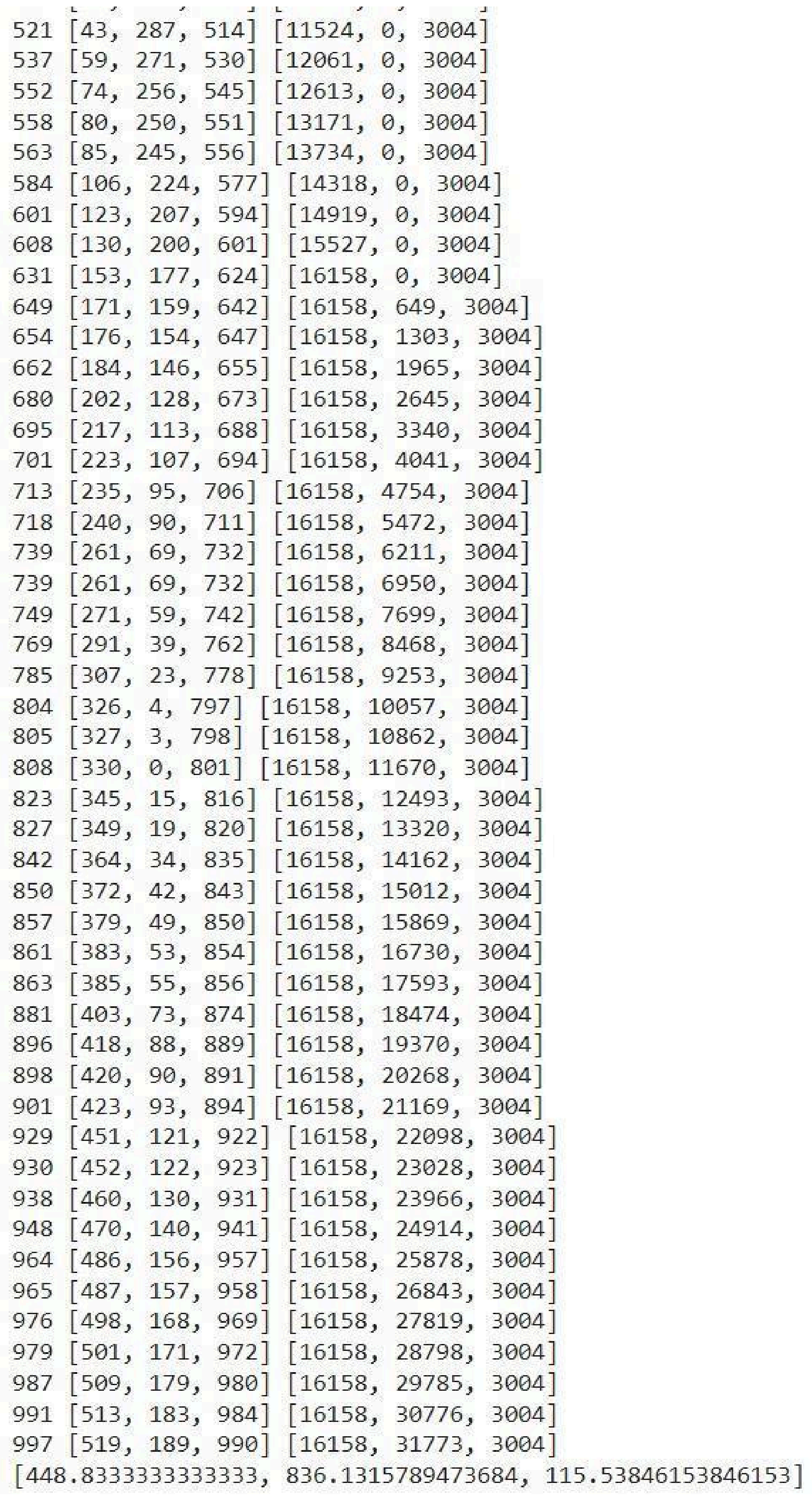


```
import random
import numpy as np
import matplotlib.pyplot as plt
#k=input("Give K:")
k=3
arr=[]
c=[]
for i in range(100):
    arr.append(random.randint(0,1000))
    c.append(i)
arr.sort()
print(arr)
mean=[]
for i in range(k): mean.append(random.choice(arr))
print(mean)
prevmean=[0]*k
while mean!=prevmean:
    sum=[0]*k
    count=[0]*k
    for ele in arr:
        distance=[0]*k
        for i in range(k):
            distance[i]=abs(ele-mean[i])

        sum[distance.index(min(distance))]+=ele
        count[distance.index(min(distance))]+=1
        print(ele, distance,sum)
    prevmean=mean
    for i in range(k):
        mean[i]=sum[i]/count[i]

print(mean)
arr=np.array(arr)
c=np.array(c)
plt.plot(arr,c,color="red")
plt.show()
```

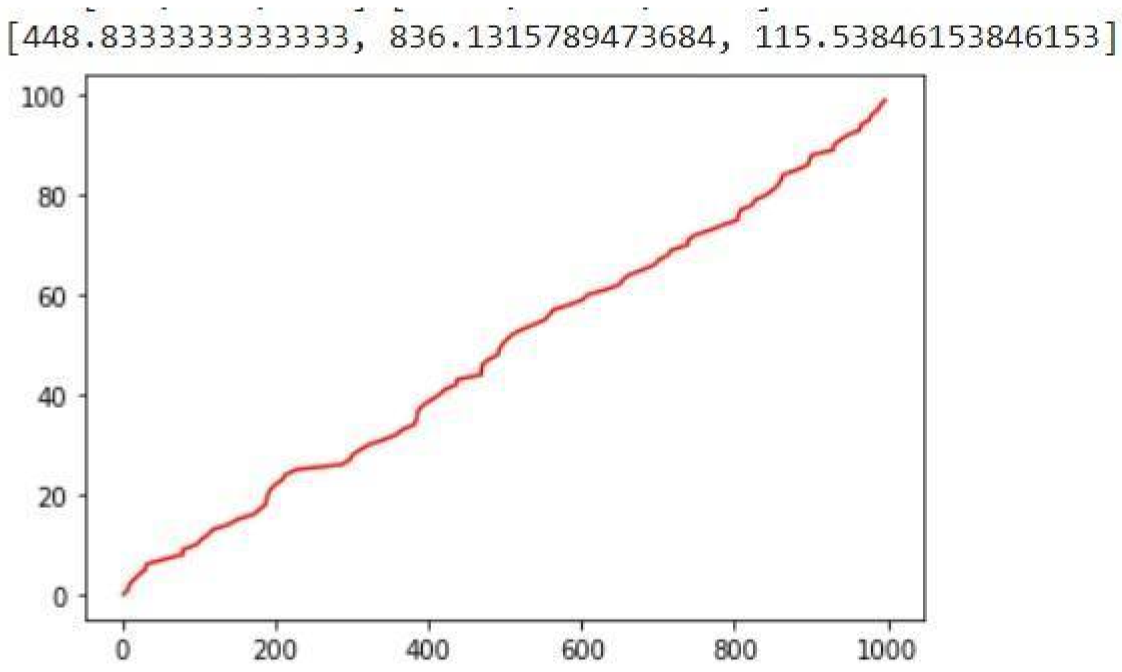
Output Clusters:



```
[1, 7, 9, 15, 22, 30, 31, 54, 79, 79, 97, 103, 112, 118, 139, 150, 171, 179, 187, 188, 190, 193, 200, 209, 213, 228, 287, 297, 301, 311, 322, 342, 358, 366, 381, 384, 385, 387, 394, 404, 414, 421, 436, 437, 469, 469, 470, 478, 491, 492, 496, 502, 509, 521, 537, 552, 558, 563, 584, 601, 608,
[478, 808, 7]
1 [477, 807, 6] [0, 0, 1]
7 [471, 801, 0] [0, 0, 8]
9 [469, 799, 2] [0, 0, 17]
15 [463, 793, 8] [0, 0, 32]
22 [456, 786, 15] [0, 0, 54]
30 [448, 778, 23] [0, 0, 84]
31 [447, 777, 24] [0, 0, 115]
54 [424, 754, 47] [0, 0, 169]
79 [399, 729, 72] [0, 0, 248]
79 [399, 729, 72] [0, 0, 327]
97 [381, 711, 90] [0, 0, 424]
103 [375, 705, 96] [0, 0, 527]
112 [366, 696, 105] [0, 0, 639]
118 [360, 690, 111] [0, 0, 757]
139 [339, 669, 132] [0, 0, 896]
150 [328, 658, 143] [0, 0, 1046]
171 [307, 637, 164] [0, 0, 1217]
179 [299, 629, 172] [0, 0, 1396]
187 [291, 621, 180] [0, 0, 1583]
188 [290, 620, 181] [0, 0, 1771]
190 [288, 618, 183] [0, 0, 1961]
193 [285, 615, 186] [0, 0, 2154]
200 [278, 608, 193] [0, 0, 2354]
209 [269, 599, 202] [0, 0, 2563]
213 [265, 595, 206] [0, 0, 2776]
228 [250, 580, 221] [0, 0, 3004]
287 [191, 521, 280] [287, 0, 3004]
297 [181, 511, 290] [584, 0, 3004]
301 [177, 507, 294] [885, 0, 3004]
311 [167, 497, 304] [1196, 0, 3004]
322 [156, 486, 315] [1518, 0, 3004]
342 [136, 466, 335] [1860, 0, 3004]
358 [120, 450, 351] [2218, 0, 3004]
366 [112, 442, 359] [2584, 0, 3004]
381 [97, 427, 374] [2965, 0, 3004]
384 [94, 424, 377] [3349, 0, 3004]
385 [93, 423, 378] [3734, 0, 3004]
387 [91, 421, 380] [4121, 0, 3004]
394 [84, 414, 387] [4515, 0, 3004]
404 [74, 404, 397] [4919, 0, 3004]
414 [64, 394, 407] [5333, 0, 3004]
421 [57, 387, 414] [5754, 0, 3004]
436 [42, 372, 429] [6190, 0, 3004]
437 [41, 371, 430] [6627, 0, 3004]
469 [9, 339, 462] [7096, 0, 3004]
469 [9, 339, 462] [7565, 0, 3004]
470 [8, 338, 463] [8035, 0, 3004]
478 [0, 330, 471] [8513, 0, 3004]
491 [13, 317, 484] [9004, 0, 3004]
492 [14, 316, 485] [9496, 0, 3004]
496 [18, 312, 489] [9992, 0, 3004]
502 [24, 306, 495] [10494, 0, 3004]
509 [31, 299, 502] [11003, 0, 3004]
```



```
521 [43, 287, 514] [11524, 0, 3004]
537 [59, 271, 530] [12061, 0, 3004]
552 [74, 256, 545] [12613, 0, 3004]
558 [80, 250, 551] [13171, 0, 3004]
563 [85, 245, 556] [13734, 0, 3004]
584 [106, 224, 577] [14318, 0, 3004]
601 [123, 207, 594] [14919, 0, 3004]
608 [130, 200, 601] [15527, 0, 3004]
631 [153, 177, 624] [16158, 0, 3004]
649 [171, 159, 642] [16158, 649, 3004]
654 [176, 154, 647] [16158, 1303, 3004]
662 [184, 146, 655] [16158, 1965, 3004]
680 [202, 128, 673] [16158, 2645, 3004]
695 [217, 113, 688] [16158, 3340, 3004]
701 [223, 107, 694] [16158, 4041, 3004]
713 [235, 95, 706] [16158, 4754, 3004]
718 [240, 90, 711] [16158, 5472, 3004]
739 [261, 69, 732] [16158, 6211, 3004]
739 [261, 69, 732] [16158, 6950, 3004]
749 [271, 59, 742] [16158, 7699, 3004]
769 [291, 39, 762] [16158, 8468, 3004]
785 [307, 23, 778] [16158, 9253, 3004]
804 [326, 4, 797] [16158, 10057, 3004]
805 [327, 3, 798] [16158, 10862, 3004]
808 [330, 0, 801] [16158, 11670, 3004]
823 [345, 15, 816] [16158, 12493, 3004]
827 [349, 19, 820] [16158, 13320, 3004]
842 [364, 34, 835] [16158, 14162, 3004]
850 [372, 42, 843] [16158, 15012, 3004]
857 [379, 49, 850] [16158, 15869, 3004]
861 [383, 53, 854] [16158, 16730, 3004]
863 [385, 55, 856] [16158, 17593, 3004]
881 [403, 73, 874] [16158, 18474, 3004]
896 [418, 88, 889] [16158, 19370, 3004]
898 [420, 90, 891] [16158, 20268, 3004]
901 [423, 93, 894] [16158, 21169, 3004]
929 [451, 121, 922] [16158, 22098, 3004]
930 [452, 122, 923] [16158, 23028, 3004]
938 [460, 130, 931] [16158, 23966, 3004]
948 [470, 140, 941] [16158, 24914, 3004]
964 [486, 156, 957] [16158, 25878, 3004]
965 [487, 157, 958] [16158, 26843, 3004]
976 [498, 168, 969] [16158, 27819, 3004]
979 [501, 171, 972] [16158, 28798, 3004]
987 [509, 179, 980] [16158, 29785, 3004]
991 [513, 183, 984] [16158, 30776, 3004]
997 [519, 189, 990] [16158, 31773, 3004]
[448.8333333333333, 836.1315789473684, 115.53846153846153]
```



```
[448.8333333333333, 836.1315789473684, 115.53846153846153]
```

B.3 Observations and learning:

In this experiment I have observed how to solve the k-mean algorithm.

B.3 Conclusion:

After completing this experiment, I learnt how to write a python code of k-mean algorithm and understood it properly.Also plot a graph about it.

****