KNN Program Testing:

K means output file for test, the format is as below:
num_items, num_attrs, number of teams assigned
item1_team_assigment, item1_attr1, item1_attr2
item2_team_assigment, item2_attr1, item2_attr2
……

```
10 2 3
0 8.000000 7.000000
1 6.000000 5.000000
2 5.000000 6.000000
1 4.000000 4.000000
0 6.000000 9.000000
1 7.000000 1.000000
2 2.000000 3.000000
2 0.000000 7.000000
1 6.000000 2.000000
0 5.000000 10.000000
```

KNN target data file, the format is as below:
num_items, num_attrs
item1_attr1, item1_attr2
item2_attr1, item2_attr2
item3_attr1, item3_attr2
item4_attr1, item4_attr2

```
4 2
4 3
2 1
1 3
3 0
```

If I input some k value that is greater than the total number of items in the output file (merely for test as shown above) from K means program, I will get an error message from the program as shown below:

```
Wuxf@DESKTOP-9NFTB1E MINGW64 ~/Dropbox/2/src/KNN/Test (master)
$ ./main.exe kmeans_out.txt knn_t.txt knn_out.txt 11
Please enter a k smaller or equal to items available.
Wuxf@DESKTOP-9NFTB1E MINGW64 ~/Dropbox/2/src/KNN/Test (master)
$
```

If I input certain k value smaller than zero, I will get another error message from the program as shown below:

```
Wuxf@DESKTOP-9NFTB1E MINGW64 ~/Dropbox/2/src/KNN/Test (master)
$ ./main.exe kmeans_out.txt knn_t.txt knn_out.txt -1
Please enter a k greater than zero.
Wuxf@DESKTOP-9NFTB1E MINGW64 ~/Dropbox/2/src/KNN/Test (master)
$
```

If I input k = 10 which is exactly the number of items in the output file for test from K means program, I will get the following outputs from the program, which is written to an output file.

```
Wuxf@DESKTOP-9NFTB1E MINGW64 ~/Dropbox/2/src/KNN/Test (master)
$ ./main.exe kmeans_out.txt knn_t.txt knn_out.txt 10 > knn_out_detail.txt

Wuxf@DESKTOP-9NFTB1E MINGW64 ~/Dropbox/2/src/KNN/Test (master)
$
```

The format of each block in the output file from my KNN program is:
(Note: itemknn denotes the items in the KNN target data file above, itemknn_i means the i^{th} KNN target data item. Itemkmeans denotes the items in Kmeans output file above.)

Data Point: i, itemknn_i_attr1, itemknn_i_attr2 Group Num: the group assignment
Squared distance, group_assignment_kmeans, itemkmeans_attr1, itemkmeans_attr2
…..
(Note: The order of the items from the Kmeans output file follows the ascending order of the distance between the item and the target data point (itemknn) analyzed by KNN)

```
Data Point: 1, 4.000000 3.000000 Group Num: 1
1.000000 1.000000 4.000000 4.000000
4.000000 2.000000 2.000000 3.000000
5.000000 1.000000 6.000000 2.000000
8.000000 1.000000 6.000000 5.000000
10.000000 2.000000 5.000000 6.000000
13.000000 1.000000 7.000000 1.000000
32.000000 0.000000 8.000000 7.000000
32.000000 2.000000 0.000000 7.000000
40.000000 0.000000 6.000000 9.000000
50.000000 0.000000 5.000000 10.000000

Data Point: 2, 2.000000 1.000000 Group Num: 1
4.000000 2.000000 2.000000 3.000000
13.000000 1.000000 4.000000 4.000000
17.000000 1.000000 6.000000 2.000000
25.000000 1.000000 7.000000 1.000000
32.000000 1.000000 6.000000 5.000000
34.000000 2.000000 5.000000 6.000000
40.000000 2.000000 0.000000 7.000000
72.000000 0.000000 8.000000 7.000000
80.000000 0.000000 6.000000 9.000000
90.000000 0.000000 5.000000 10.000000

Data Point: 3, 1.000000 3.000000 Group Num: 1
1.000000 2.000000 2.000000 3.000000
10.000000 1.000000 4.000000 4.000000
17.000000 2.000000 0.000000 7.000000
25.000000 2.000000 5.000000 6.000000
26.000000 1.000000 6.000000 2.000000
29.000000 1.000000 6.000000 5.000000
40.000000 1.000000 7.000000 1.000000
61.000000 0.000000 6.000000 9.000000
65.000000 0.000000 8.000000 7.000000
65.000000 0.000000 5.000000 10.000000

Data Point: 4, 3.000000 0.000000 Group Num: 1
10.000000 2.000000 2.000000 3.000000
13.000000 1.000000 6.000000 2.000000
17.000000 1.000000 4.000000 4.000000
17.000000 1.000000 7.000000 1.000000
34.000000 1.000000 6.000000 5.000000
40.000000 2.000000 5.000000 6.000000
58.000000 2.000000 0.000000 7.000000
74.000000 0.000000 8.000000 7.000000
90.000000 0.000000 6.000000 9.000000
104.000000 0.000000 5.000000 10.000000
```

K means output file for test, the format is as below:
num_items, num_attrs, number of teams assigned
item1_team_assigment, item1_attr1, item1_attr2
item2_team_assigment, item2_attr1, item2_attr2
……

```
10 2 3
0 8.000000 7.000000
1 6.000000 5.000000
2 5.000000 6.000000
1 4.000000 4.000000
0 6.000000 9.000000
1 7.000000 1.000000
2 2.000000 3.000000
2 0.000000 7.000000
1 6.000000 2.000000
0 5.000000 10.000000
```

KNN target data file, the format is as below:
num_items, num_attrs
item1_attr1, item1_attr2
item2_attr1, item2_attr2
item3_attr1, item3_attr2
item4_attr1, item4_attr2

```
4 2
4 3
2 1
1 3
3 0
```

The above files show that my KNN program is running correctly.
For example, if we pick the 4th data point in KNN target data which is (3,0), its squared distance to point (2,3) in Kmeans output is 10, as recorded in the first column, and the distance is shorter than the other distances from other Kmeans data points to this output point. Also, for this target data point (3,0), the first kth (k = 10) closest Kmeans data points consist of 4 from group 1, 3 from group 2 and 3 from group 0. As a result, its group assignment should be group 1, which is already written after "Group Num:"

Data Analysis for kmeans and knn:

Groups:

Num in Group: 66
Total Distance Squared: 219.895343
Centroid: 4.342621 79.439394

Num in Group: 58
Total Distance Squared: 577.545772
Centroid: 2.006362 51.155172

Num in Group: 38
Total Distance Squared: 230.039757
Centroid: 4.100368 73.315789

Num in Group: 35
Total Distance Squared: 301.114976
Centroid: 2.251829 61.285714

Num in Group: 53
Total Distance Squared: 588.738350
Centroid: 4.384566 86.509434

On the left is the output results generated by Kmeans by setting k equal to 5 (i.e. in total we created 5 groups). We think that k = 5 is optimal for Kmeans since the number of data points (items) assigned to each group is more close to each other. (i.e. the data points are more averagely distributed.)
This makes sense because each different centroid of tuple with duration time and time since last erupted has close number of group members. The result from the numerical analysis is realistic because in real time the Geyser eruption cases should be averagely distributed to different categories with different mean eruption duration and mean inter-eruption time.

1 2.200000 54.000000
4 4.450000 83.000000
2 3.567000 73.000000
2 4.500000 73.000000
4 4.150000 88.000000
0 3.817000 80.000000
2 3.917000 71.000000
4 4.450000 83.000000
1 2.000000 56.000000
0 4.283000 79.000000
0 4.767000 78.000000
4 4.533000 84.000000

On the left is the output results generated by KNN by setting k equal to 10 (i.e. select 10 closest data points from Kmeans output when assigning group number to each target data point)
This makes sense because the number of target data points classified to each group is also close to each other, which corresponds to the output generated by Kmeans above.

3 1.850000 58.000000
4 4.250000 83.000000
1 1.983000 43.000000
3 2.250000 60.000000
2 4.750000 75.000000
0 4.117000 81.000000
1 2.150000 46.000000
4 4.417000 90.000000
1 1.817000 46.000000
2 4.467000 74.000000