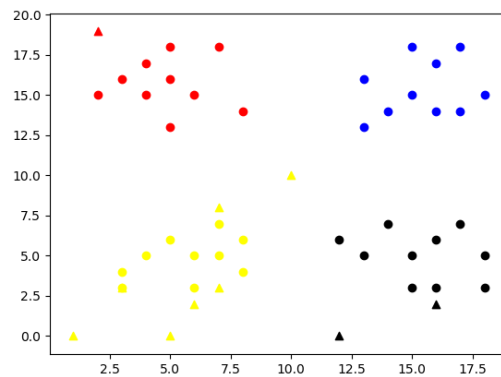
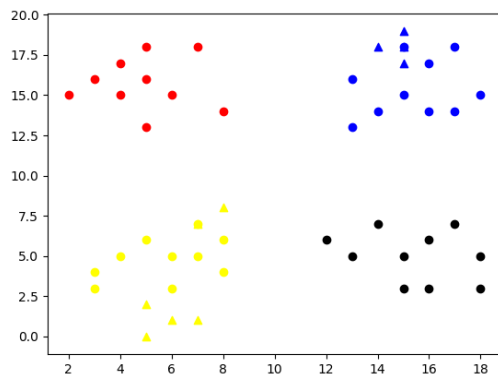


Xin Yang  
NetID: xy213

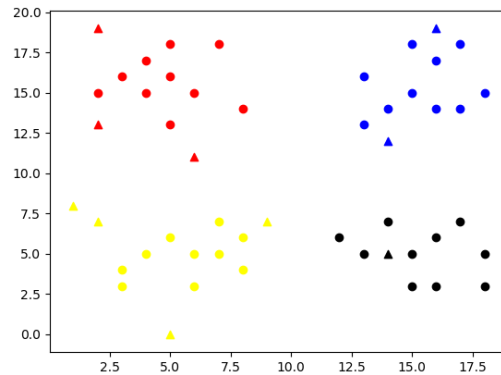
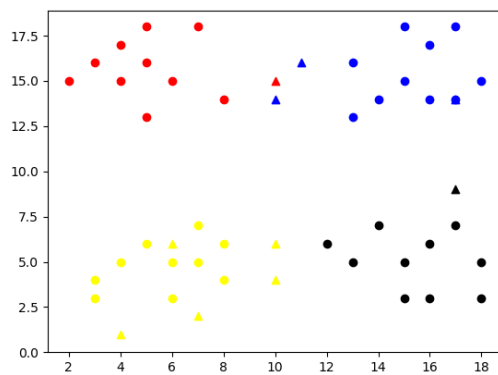
## Homework 1 - Programming Part

Problem 2:

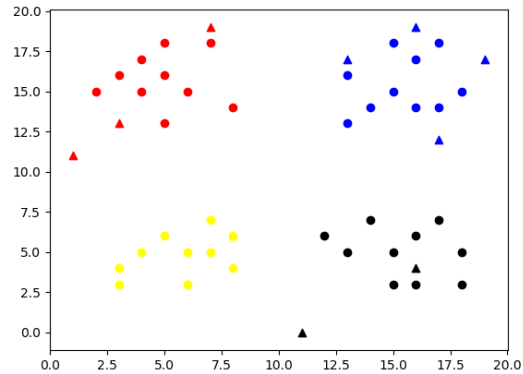
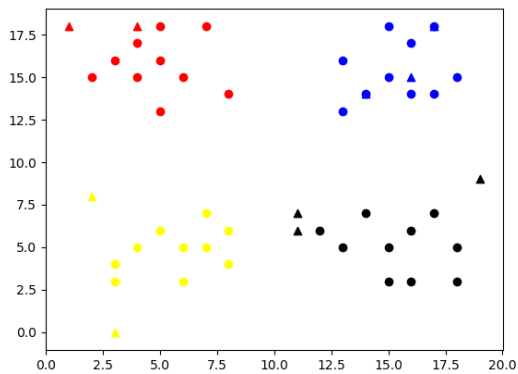
K=3:



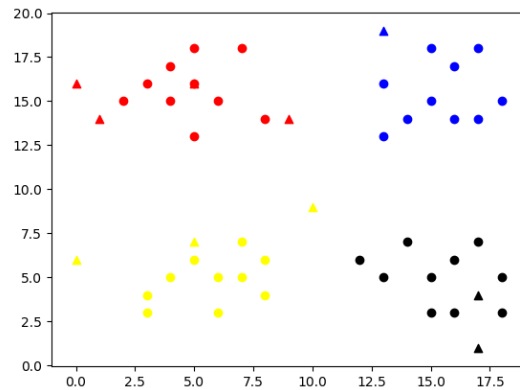
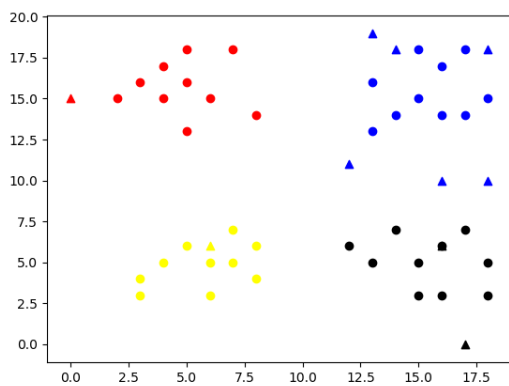
K=4:



K=5:



K=10:



Two screenshots of the problem 2 with different Ks(3, 4, 5, 10) are shown as above. The program can correctly classify the test data(triangle points) to the corresponding clusters. In the program, I manually converted all the data to signed integer and then calculated the L2 distance. For some test cases in the center of multiple clusters, a bigger K value can usually give us a more fair result, while a smaller K might be “short sighted” and misclassify some points in extreme cases.

Problem 3:

For this problem, I also converted all input data to signed integer, then I tried two size of test data: 20 images and 200 images. For each test data size, I tried different K values:3, 5, and 10. The program can classify images in a good result as shown below.

For 20 images with K=3:

```
---classification accuracy for knn on mnist: 1.0 ---
---execution time: 13.321259021759033 seconds ---
```

For 20 images with K=5:

```
---classification accuracy for knn on mnist: 1.0 ---  
---execution time: 14.177934169769287 seconds ---
```

For 20 images with K=10:

```
---classification accuracy for knn on mnist: 1.0 ---  
---execution time: 14.559323787689209 seconds ---
```

For 200 images with K=3:

```
---classification accuracy for knn on mnist: 0.985 ---  
---execution time: 133.58575296401978 seconds ---
```

For 200 images with K=5:

```
---classification accuracy for knn on mnist: 0.985 ---  
---execution time: 134.819895029068 seconds ---
```

For 200 images with K=10:

```
---classification accuracy for knn on mnist: 0.975 ---  
---execution time: 134.68297719955444 seconds ---
```