Q3

(1.) For the implementation of the k-medoids framework I chose the initial medoids randomly. Then I calculated the distance from every data point to the given medoids. Using the euclidean distance $d(p,q) = \sqrt{(p-q)^2}$ but also manhattan distance $d(p,q) = \sum_{i=1}^{n} |p_i - q_i|$

Afterwords the data points where assigned to the cluster with the least distance.

Following the medoids where adjusted. ~~that~~ ~~it is~~

The center for every cluster was updated with the data point which was closest to the centroid of ~~the~~ each cluster. These steps were repeated for a certain amount of iterations or when the distortion function didn't decrease. Although, I mainly focused on the max iteration since the distortion function was causing performance issues

(2) Depending on the number of the Clusters k, a different amount of colours where displayed in the compressed pictures. ~~This also resulted in~~ The more clusters the more iterations until ~~con~~ convergence. The observation for different k where the following:

# Cluster	2	3	4	7
# Iteration	10	13	19	>30 *

* runtime error manually stopped

③ Differences in the pictures based on different initial centroids were not significant. However, the run time for the clustering varied based on the different initial starts. When choosing a poor assignment as initial centroids (e.g. really close together) I was observing a considerable increase in terms of iterations and run time.

④ In general the solutions given through the k-means algo dont differ significantly from k-medoids. However, for smaller cluster values it appears that k-medoid is more effective capturing the seperation of the colours. The running time was quite similar to k-medoids with a slight increase for the football picture (since its bigger). This was quite surprising since k-medoids is supposed to be much faster.