Question 1

Yufei Yin

Question 1

- In this question, you will implement kmedians with L1 distances.
- Write an R function kmedians that takes a numeric data frame x (with one row per data item, and one column per dimension D), and a positive integer K (for the number of clusters), and a positive integer iters for the number of iterations.
- The function should be a modification of the kmeans algorithm, except:
  1) In the step where the cluster locations are updated, set the cluster locations to be the medians of the data items assigned to the cluster (by setting each dimension of the cluster location to the median of the values for that dimension of the data items assigned to that cluster).
  2) In the step where data items are assigned to the nearest cluster, assign them to the nearest cluster based on their L1 distance to the cluster locations (instead of their Euclidean distance).
- Your function should return a list with an element named "locations" providing a K x D table showing the cluster locations, and "assignments" providing a vector of length N showing the cluster assignments at the last iteration. Provide your code.

```r
kmedians = function(x, k, iters){
  N = dim(x)[1]
  D = dim(x)[2]

  centres = matrix(NA, k, D)
  clusters = rep(NA, N)

  for (i in 1:N){
    clusters[i] = sample.int(k, 1)
  }

  for (iter in 1:iters){
    for (k in 1:k){
      for (d in 1:D){
        centres[k, d] = median(x[clusters == k, d])
      }
    }
    distanceMatrix <- matrix(NA, nrow=N, ncol=k)
    for(i in 1:k) {
      distanceMatrix[,i] <- rowSums(t(abs(t(x)-centres[i,])))
    }
    clusters <- apply(distanceMatrix, 1, which.min)
    centres <- apply(x, 2, tapply, clusters, median)
  }
  return(list(locations=centres, assignment=clusters))
}
```