Goals
A commonly used algorithm to perform clustering is k-means. The goal of this tutorial is to implement k-means and apply it to a small data set.

Problem setting
The MATLAB-function `toydata(p,n,verbose)` (see `toydata.m`) generates a small 2-dimensional data set which contains `p` clusters. In this tutorial we use the data generated with `toydata(p,n,verbose)` and apply the k-means-algorithm to this data.

Task 1
Write the following MATLAB-function

```matlab
function R = findMapping(X,Means)
```

which returns the mapping `R` of the data `X` for given cluster centers `Means` (expectation step of k-means). Here `X` is a `n × m` matrix of `n` instances with `m` numerical attributes. `Means` is a `k × m` matrix with `k m`-dimensional cluster centers. The returned matrix `R ∈ R^{n×k}` is defined as follows: \( R_{ij} = 1 \) if the `i`-th instance belongs to cluster `j`, \( R_{ij} = 0 \), otherwise.

Task 2
Write the following MATLAB-function

```matlab
function Means = findMeans(X,R)
```

which computes the cluster centers for a given mapping `R` of the data `X` (maximization step of k-means). Here `X`, `Means` and `R` are defined as described in Task 1.
Task 3
Write the following MATLAB-function

\[
\text{function } y = \text{kmeans}(X,k)
\]

which identifies \( k \) clusters using the k-means algorithm. The data matrix \( X \) is defined as described in Task 1. The returned vector \( y \) contains for each data point the the cluster numbers \( \in \{1,\ldots,k\} \) it belonging to.

Task 4
Create different data sets using the MATLAB-function \( X = \text{toydata}(p) \) for varying values for \( p \). Investigate the performance of k-means for a different number of clusters \( k < p \), \( k = p \) und \( k > p \). Try to visualize the data; the first cluster can be plotted using the MATLAB-call \( \text{plot}(X(Y==1,1), X(Y==1,2), '+') \).