On a donc au départ p variables X1, X2, ..., Xp et n individus. La composante j de l'individu i est notée Xi,j et se trouve à la j-ème colonne de la i-ème ligne.

Algorithme du k-means :

Paramètres en entrée : Matrice X(n,p), entier naturel k
Paramètres en sortie : Matrice M(n,p+1) où la dernière colonne traduit l'affectation du vecteur individu à un des k clusters

Début

```
// 1ère itération
// sélectionner k individus (au hasard par ex) à partir de la matrice X
// ces k individus sont les représentants des k clusters
!!! MANQUE !!!

CheckChange <-- (true,true,...,true) (vecteur de n lignes)

// création et remplissage de la matrice M contenant nos données
// et l'attribution d'un individu à un cluster (dernière colonne)
M = matrice(n,p+1)

Pour i=0 à n
        Pour j=0 à p
                M(i,j) = X(i,j)
        Fin pour
Fin pour
Pour i=0 à n
        M(i,p+1)=0
Fin pour

// matrice des centres de gravité des k clusters
K = matrice(k,p)

Tant que (CheckChange != (false,false,...,false))

        // affecter les n individus aux k clusters :
        Pour i=0 à n-1

                VecteurDistance <-- (0,0)
                distance <-- 0
                OldCluster <-- -1

                Pour j=0 à k-1
                        // calcul de la distance euclidienne entre l'individu X(j)
                        // et le barycentre du cluster j K(j)
                        distance <-- |X(j)-K(j)|

                        // on va voir si la nouvelle distance est plus petite
                        // que la plus petite de toutes les distances précédentes
                        Si j=1 alors
                                VecteurDistance(x) = distance
                                VecteurDistance(y) = j
                        Sinon si ((distance) < VecteurDistance(x)) alors
                                VecteurDistance(x) = distance
                                VecteurDistance(y) = j
                        Fin si
                Fin pour

                // attribuer l'individu à son (nouveau) cluster
                // i.e. ajouter à la dernière colonne de la matrice M l'indice du cluster
                OldCluster <-- M(i,p+1)
                M(i,p+1) <-- VecteurDistance(y)
```

```
            // Regarder si l'indice du cluster a changé
            // Si oui, CheckChange(i) = true I Si non, CheckChange(i) = false
            Boolean CheckChange(i) <-- isDifferent(Oldcluster,VecteurDistance(y))

    Fin pour

    // calcul des k nouveaux centres de gravité
    Pour j=0 à k-1
        Nbindividus <-- 0
        Somme <-- (0,0,...,0) //(1 ligne, p colonnes)
        Pour i=0 à n-1
            Si (M(i,p+1) = j) alors
                Pour l=0 à p-1
                    Somme(l) = Somme(l) + M(i,l)
                    NbIndividus++;
                Fin pour
            Fin si
        Fin pour
        Pour l=0 à p-1
            K(j,l) <-- Somme(l) / NbIndividus
        Fin pour
    Fin pour

Fin Tant que

Fin
```