public class KMeansClustering {

public static void main(String args[]) {

int arr[] = {2, 4, 10, 12, 3, 20, 30, 11, 25}; // initial data

int i, m1, m2, a, b, n = 0;

boolean flag;

float sum1, sum2;

a = arr[0];

b = arr[1];

m1 = a;

m2 = b;

int cluster1[] = new int[arr.length], cluster2[] = new int[arr.length];

do {

sum1 = 0;

sum2 = 0;

cluster1 = new int[arr.length];

cluster2 = new int[arr.length];

n++;

int k = 0, j = 0;

for (i = 0; i < arr.length; i++) {

if (Math.abs(arr[i] - m1) <= Math.abs(arr[i] - m2)) {

cluster1[k] = arr[i];

k++;}

else {

cluster2[j] = arr[i];

j++;} }

System.out.println();

for (i = 0; i < k; i++) {

sum1 = sum1 + cluster1[i] }

for (i = 0; i < j; i++) {

sum2 = sum2 + cluster2[i]; }

//printing Centroids/Means\

System.out.println("m1=" + m1 + " m2=" + m2);

a = m1;

b = m2;

m1 = Math.round(sum1 / k);

m2 = Math.round(sum2 / j);

flag = !(m1 == a && m2 == b);

System.out.println("After iteration " + n + " , cluster 1 :\n"); //printing the clusters of each iteration

for (i = 0; i < cluster1.length; i++) {

System.out.print(cluster1[i] + "\t"); }

System.out.println("\n");

System.out.println("After iteration " + n + " , cluster 2 :\n");

for (i = 0; i < cluster2.length; i++) {

System.out.print(cluster2[i] + "\t");

} }

while (flag);

System.out.println("Final cluster 1 :\n"); // final clusters

for (i = 0; i < cluster1.length; i++) {

System.out.print(cluster1[i] + "\t"); }

System.out.println();

System.out.println("Final cluster 2 :\n");

for (i = 0; i < cluster2.length; i++) {

System.out.print(cluster2[i] + "\t");

}}}

//output

Z:\dbmi>javac KMeansClustering.java

Z:\dbmi>java KMeansClustering

m1=2 m2=4

After iteration 1 , cluster 1 :

2 3 0 0 0 0 0 0 0

After iteration 1 , cluster 2 :

4 10 12 20 30 11 25 0 0

m1=3 m2=16

After iteration 2 , cluster 1 :

2 4 3 0 0 0 0 0 0

After iteration 2 , cluster 2 :

10 12 20 30 11 25 0 0 0

m1=3 m2=18

After iteration 3 , cluster 1 :

2 4 10 3 0 0 0 0 0

After iteration 3 , cluster 2 :

12 20 30 11 25 0 0 0 0

m1=5 m2=20

After iteration 4 , cluster 1 :

2 4 10 12 3 11 0 0 0

After iteration 4 , cluster 2 :

20 30 25 0 0 0 0 0 0

m1=7 m2=25

After iteration 5 , cluster 1 :

2 4 10 12 3 11 0 0 0

After iteration 5 , cluster 2 :

20 30 25 0 0 0 0 0 0

uster 1 :

2 4 10 12 3 11 0 0 0

Final cluster 2 :

20 30 25 0 0 0 0 0 0