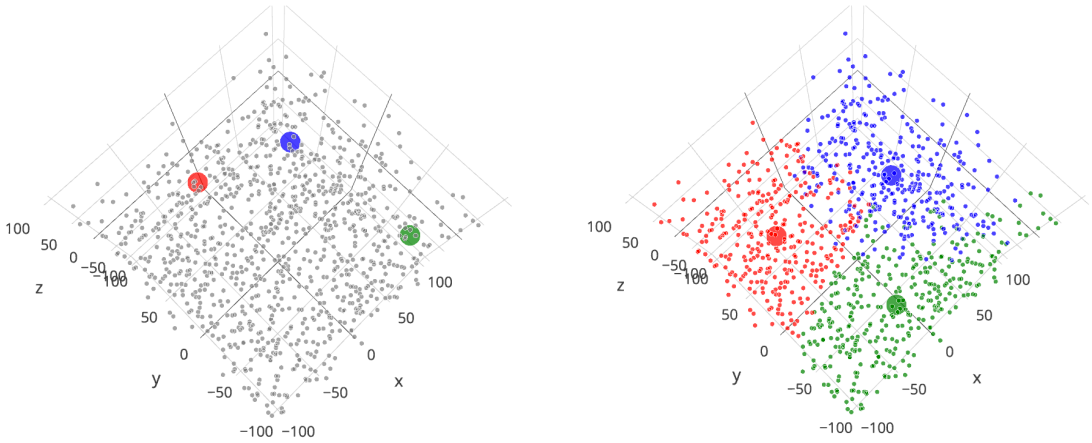
Problem D

K-Means Clustering

K-Means clustering is a method that allows the modeling of probability density functions by the distribution of prototype vectors and is popular for cluster analysis in data mining. The method will partition a set of *n* observations (points) into *k* clusters, where each point will be associated with the cluster with the nearest mean. The Euclidean distance is usually the adopted metric for proximity.

Clustering is a NP-hard problem, but there are efficient heuristic algorithms that can quickly to a local optimum. In this implementation, the application takes as input the coordinates (3D) of *k* initial centroids and a set of data points. K-Means performs an iterative process, where points are re-clustered according to the minimum Euclidean distance between them and the centroids. Next, the centroid of each partition is recalculated taking the mean of all points in the partition, and the whole procedure is repeated until no centroid is changed and no points are assigned to another cluster. Upon completion, the algorithm returns the coordinates of the final *k* centroids.



(a) initial centroids (b) re-clustered centroids

Figure D1. K-Means example with 3 centroids and 1,000 points on $\mathbb{R}^3$.

Figure D1 provides an example of K-Means in with 3 centroids and 1,000 points on $\mathbb{R}^3$. Figure D1(a) shows the initial position of the centroids (big points in red, green and blue), while Figure D1(b) shows the final position of the centroids and points associated with each cluster (with the same color of their respective centroids).

Write a parallel version of the given sequential program. We are not interested in other clustering methods; hence you should not change this aspect of the application.

Input

The first two lines with contain the number of centroids *k* and the number of points *n*. The next *k*+*n* lines will contain (*x*, *y*, *z*) coordinates for *k* centroids and *n* points, respectively.
The input must be read from the standard input.

Output

The program will print *k* lines containing (*x*, *y*, *z*) coordinates of the re-clustered *k* centroids.
The output must be written to the standard output.

Example

Input	Output for the input
3 9 -2.0 0.0 1.5 1.0 1.0 1.0 -1.0 -1.0 0.0 -1.2 0.6 0.8 1.5 0.5 1.8 0.5 -1.0 0.8 -0.5 1.2 0.7 1.0 1.5 -0.8 -1.2 0.2 0.5 0.2 1.0 0.2 -0.2 -1.0 1.8 0.5 -1.5 1.5	-0.97 0.67 0.67 0.90 1.00 0.40 0.27 -1.17 1.37