Template for the function vqlbg.m

```
function c = vqlbg(d, k)

% VQLBG Vector quantization using the Linde-Buzo-Gray algorithm

%

% Inputs:

%       d contains training data vectors (one per column)

%       k is number of centroids required

%

% Outputs:

%       c contains the result VQ codebook (k columns, one for each centroids)
```

Hints

Cluster Vectors (Nearest-Neighbor Search):

The nearest-neighbor search step is: given a current codebook **`c`**, assign each training vector in **`d`** with a closest codeword. To do that, one needs to compute the pair-wise distances between each vectors in **`d`** to each vectors (codeword) in **`c`**. This can be done with the supplied function `disteu`:

```
z = disteu(d, c);
```

Now `z(i, j)` would be the distance between the training vector `d(:, i)` and the codeword `c(:, j)`. Next step, for each training vector, find the closest codeword. For this, use the Matlab function `min`:

```
[m, ind] = min(z, [], 2);
```

The result index vector `ind` contains the associated cluster number for each training vector. So to access to all the training vectors that belong to the cluster number `j` (those vectors that are closest to the codeword `c(:, j)`), one can use:

```
d(:, find(ind == j));
```

Find Centroids

The centroid of all vectors in a particular cluster is found by the Matlab function `mean`. For example, after the Nearest-Neighbor Search step above, the new centroids of the clusters number `j` can be updated as follows:

```
c(:, j) = mean(d(:, find(ind == j)), 2);
```