MATLAB Programming NCTU Spring 2017 LAB#14 2017/6/9

Main Task (C/C++ MEX function):

In this lab you will try to code a function in C, compile it into a MEX file, and then use within MATLAB.

The function you will implement is called **getPairD**. It has the form **D=getPairD(A,B)**, where **A** and **B** are 2-D matrices representing sets of point coordinates. Specifically, each column in **A** and **B** represents the coordinates of a point.

The matrices **A** and **B** should have the same number of rows, which is the number of dimensions of the points coordinates.

The output **D** is a 2-D matrix containing pairwise distances between the point sets **A** and **B**.

See the right for an example.

Argument checking:

- There are two inputs and one output.
- Both inputs are 2-D double arrays.
- The two inputs have the same number of rows.

```
>> A = [0 0; 1 3; 2 4]'

A =

     0     1     2
     0     3     4

>> B = [1 0; 0 1; 2 0; 2 1]'

B =

     1     0     2     2
     0     1     0     1

>> D = getPairD(A, B)

D =

    1.0000    1.0000    2.0000    2.2361
    3.0000    2.2361    3.1623    2.2361
    4.1231    3.6056    4.0000    3.0000
```

The C part of the code for you to use:

```c
#include <math.h>

// L: dimension (#rows) of the input points
// N1 and N2: #points in V1 and V2
// V1 and V2: inputs points (treated as vectors in MATLAB linear indexing order)
// D: output matrix (treated as vector in MATLAB linear indexing order)
void get_pair_d(int L, int N1, int N2, const double *V1, const double *V2, double *D)
{
  int i, j, q1, q2, iL;
  int k = 0; // for linear index into D
  double dv;
  for (i = 0;  i < N2;  i++) {
    q1 = i * L; // linear index offest in V2
    for (j = 0;  j<N1;  j++) {
      q2 = j * L; // linear index offest in V1
      D[k] = 0;
      for (iL = 0;  iL < L;  iL++) {
        dv = V2[q1+iL] - V1[q2+iL];
        D[k] += dv * dv;
      }
      D[k] = sqrt(D[k]);
      k++;
    }
  }
}
```