ANL Tutorial Exercise

Parallel Key-Value Store-Retrieval

In this tutorial we will learn:

- invoking remote entry methods on chare array elements, and
- doing reductions in chare arrays

This example has two chare arrays - KeyValueStore and KeyValueClient

1. Each element of KeyValueStore array, stores pairs for keys in the range [idx*M, (idx+1)*M-1].
2. KeyValueClient array has a method called run(), which generates K random keys and then requests for the values of those keys from their respective owners.
3. Once all the values of the requested keys have been received, the chare array elements contributes to a reduction indicating that it is done.
4. The reduction target is a function that terminates the program.

Note that even though each client chare knows how many requests it is making, the keyValueStore element chares do not know how many requests they will each get.
This could pose some problems in MPI; but its OK in Charm++. When you contribute into the reduction, the keyValueStore chares on your processor can continue responding to requests.

Begin with incomplete code below (and can be downloaded here).

```
rcise
eduction, sdag




inProxy;
eStore kvstoreProxy;
eClient kvclientProxy;




;
 void finish();


nt {

 refnum, int value);
```

```
          add code here
          1. send the request for values for all the keys in the vector kvpairs
                   use method KeyValueStore::request(..) for this
          2. the corresponding array elements of KeyValueStore Array will send
                   the values back by calling the KeyValueClient::response(..) that will
                   also have the request refnum.
                   save the value at the correct index (by using refnum) of the kvpairs vector
          3. once all the responses have been received, do a reduction with Main::finish() as
             the reduction target  which will exit the program
  *********/
```

```
e {
d);
refnum, int k, int reqIdx);
```

The solution can be found here.