Advanced Programming Techniques (java or c #) 2

Linq

1. Given a set of numbers, write a queries that
   a. Select numbers greater than 80
   b. Order numbers, descending
   c. Transforms numbers into given string: “Have number #n”, where #n is replaced by the number

   For a, b and c write queries in both query syntax and method syntax.

   d. Counts number that are smaller than 100 and greater than 70. Use query syntax for selecting, store query in a variable and use method for counting.
2. Given a set of persons, write queries that
   a. Select persons with particular height (wrap it in function)
   b. Transform name and last name into format -> John Doe into J. Doe
   c. Person has optional list of allergies, separated by a coma. Select all distinct allergies (make sure every allergy appears only once).
   d. Select number of cities that start with letter “H”.
   e. Join list of cities with list of persons on city name. Select persons from cities with population above 100 000.

   For a and b and c write queries in both query syntax and method syntax.

   f. Manually add a list with three different city names. Use it to write two queries, one that selects persons that live in those cities, and one that selects persons that don’t live in those cities.
3. Using LINQ and XElement class, convert persons list to XML.

```
int[] numbers =
[106,104,10,5,117,174,95,61,74,145,77,95,72,59,114,95,61,116,106,66,75,85,1
04,62,76,87,70,17,141,39,199,91,37,139,88,84,15,166,118,54,42,123,53,183,95
,101,112,26,41,135,70,48,59,69,109,93,110,153,178,117,5];

City[] cities = [

    new City("Toronto", 100200),

    new City("Hamilton", 80923),

    new City("Ancaster", 4039),
```

```
            new City("Brantford", 500890),
            ];

Person[] persons = [
  new Person("Cedric","Coltrane","Toronto",157,null),
  new Person("Hank","Spencer","Peterborough",158,"Sulfa, Penicillin"),
  new Person("Sara","di","29",145,null),
  new Person("Daphne ","Seabright","Ancaster",146,null),
  new Person("Rick","Bennett","Ancaster",220,null),
  new Person("Amy","Leela","Hamilton",172,"Penicillin"),
  new Person("Woody","Bashir","Barrie",153,null),
  new Person("Tom", "Halliwell","Hamilton",179,"Codeine, Sulfa"),
  new Person("Rachel ","Winterbourne","Hamilton",163,null),
  new Person("John","West","Oakville",138,null),
  new Person("Jon","Doggett","Hamilton",194,"Peanut Oil"),
  new Person("Angel","Edwards","Brantford",176,null),
  new Person("Brodie","Beck","Carlisle",157,null),
  new Person("Beanie","Foster","Ancaster",154,"Ragweed, Codeine"),
  new Person("Nino","Andrews","Hamilton",186,null),
  new Person("John","Farley","Hamilton",213,null),
  new Person("Nea","Kobayakawa","Toronto",147,null),
  new Person("Laura","Halliwell","Brantford",146,null),
  new Person("Lucille","Maureen","Hamilton",184,null),
  new Person("Jim","Thoma","Ottawa",173,null),
  new Person("Roderick","Payne","Halifax",58,null),
  new Person("Sam","Threep","Hamilton",199,null),
  new Person("Bertha","Crowley","Delhi",125,"Peanuts, Gluten"),
  new Person("Roland","Edge","Brantford",199,null),
```

```
 new Person("Don","Wiggum","Hamilton",189,null),
 new Person("Anthony","Maxwell","Oakville",92,null),
 new Person("James","Sullivan","Delhi",139,null),
 new Person("Anne","Marlowe","Pickering",165,"Peanut Oil"),
 new Person("Kelly","Hamilton","Stoney",84,null),
 new Person("Charles","Andonuts","Hamilton",62,null),
 new Person("Temple ","Russert","Hamilton",166,"Sulphur"),
 new Person("Don","Edwards","Hamilton",215,null),
 new Person("Alice","Donovan","Hamilton",167,null),
 new Person("Stone","Cutting","Hamilton",110,null),
 new Person("Neil","Allan","Cambridge",203,null),
 new Person("Cross","Gordon","Ancaster",125,null),
 new Person("Phoebe","Bigelow","Thunder",183,null),
 new Person("Harry","Kuramitsu","Hamilton",210, null)
];
```