1. Using LINQ, write a function that returns all words that contain the substring “th” from a list.

* List<string> words = new List<string>() { "the", "bike", "this", "it", "tenth", “mathematics” };

2. Using LINQ, write a function that takes in a list of strings and returns a copy of the list without duplicates.

* List<string> names = new List<string>() { "Mike", "Brad", "Nevin", "Ian", "Mike" };

3. Using LINQ, write a function that calculates the class grade average after dropping the lowest grade for each student. The function should take in a list of strings of grades (e.g., one string might be "90,100,82,89,55"), drops the lowest grade from each string, averages the rest of the grades from that string, then averages the averages.

* List<string> classGrades = new List<string>()

{

"80,100,92,89,65",

"93,81,78,84,69",

"73,88,83,99,64",

"98,100,66,74,55"

};

* Expected output: 86.125

4. Write a function that takes in a string of letters (i.e. “Terrill”) and returns an alphabetically ordered string corresponding to the letter frequency (i.e. "E1I1L2R2T1")

SQL = SELECT, FROM, WHERE, ORDERBY

LINQ = Where, OrderBy, Select

Take string

Turn into a list

Sort alphabetically

Search list to see if how many times the character is repeat

Print the letter

Print the Count

1. Using LINQ, write a function that returns all words that contain the substring “th” from a list.

* List<string> words = new List<string>() { "the", "bike", "this", "it", "tenth", “mathematics” };

Take in string

Check to see if word contains a th

Delete word

Continue

2. Using LINQ, write a function that takes in a list of strings and returns a copy of the list without duplicates.

* List<string> names = new List<string>() { "Mike", "Brad", "Nevin", "Ian", "Mike" };

3. Using LINQ, write a function that calculates the class grade average after dropping the lowest grade for each student. The function should take in a list of strings of grades (e.g., one string might be "90,100,82,89,55"), drops the lowest grade from each string, averages the rest of the grades from that string, then averages the averages.

* List<string> classGrades = new List<string>()

{

"80,100,92,89,65",

"93,81,78,84,69",

"73,88,83,99,64",

"98,100,66,74,55"

};

* Expected output: 86.125

Loop through every list

Turn string into a list

Turn String List into an Int List

Remove Lowest Int from List

Get Average From List

Get average of the averages.