**C# Linq Hands on Assignment**

**Program 1:**

Oracle and Microsoft has a list of CRM products. A Product object has the following properties:

ProductCompany, ProductCategory,ProductName & ProductPrice

Write a Program using C# linq to find the prices of Microsoft CRM products that are less than Oracle CRM Products.

Input should be a hard coded list of Product objects from the below sample file. Each file has products from Microsoft and Oracle.

**Hint:** Use compound “from” clause - Filtering

**Sample Input:**

Sample Microsoft & Oracle Products:

****

**Sample Output:** To be printed on console

Microsoft CRM Product Navision is less in price than Oracle CRM Product O-CRM

Microsoft CRM Product Navision is less in price than Oracle CRM Product Cloud-CRM

Microsoft CRM Product Navision is less in price than Oracle CRM Product On-Premise-CRM

Microsoft CRM Product Axapta is less in price than Oracle CRM Product On-Premise-CRM

Microsoft CRM Product Dynamic AX is less in price than Oracle CRM Product On-Premise-CRM

Microsoft CRM Product NetSuite is less in price than Oracle CRM Product On-Premise-CRM

**Sample Solution:** TBD

**Program 2:**

Given a list of employees in a file, write a program to group the employees by first letter of their FirstName property. Each employee object has the 3 properties: ID, FirstName & Surname

Input should be hard coded list of Employee objects from the below sample file

**Sample Input File:**

**Hint:** Use Group By clause



**Sample Output:** To be printed on console

Employees whose name starts with letter "A":

FirstName: Adam SurName: David

FirstName: Aditya SurName: Jalgaonkar

FirstName: Akshaya SurName: Kishore

Employees whose name starts with letter "S":

FirstName: Subhash SurName: Chatterjee

FirstName: Soumitra SurName: Panda

Employees whose name starts with letter "K":

FirstName: Kay SurName: Jenkins

FirstName: Kaushik SurName: Mitra

Employees whose name starts with letter "M":

FirstName: Mandy SurName: Bagwell

FirstName: Mandana SurName: Banaerjee

Employee whose name starts with letter "P":

FirstName: Piyush SurName: Bhatnagar

**Sample Solution:** TBD

**Program 3:**

Write a program to reverse employee first names in the list of employees whose second letter of SurName property is any vowel. Each employee object has the 3 properties: ID, FirstName & Surname

Input should be a hard coded list of Employee objects from the below sample file

**Hint**: Use Sorting

**Sample Input of Employee List:**

****

**Sample Output:** To be printed on console

Reverse names of Employee First Names whose second letter of SurName is a vowel:

FirstName: madA SurName: David

FirstName: yaK SurName : Jenkins

FirstName: ydnaM SurName: Bagwell

FirstName: artimuoS SurName: Panda

FirstName: Kihsuak SurName: Mitra

FirstName: anadnaM SurName: Banaerjee

FirstName: Aytida SurName: Jalgaonkar

FirstName: Ayahska SurName: Kishore

**Sample Solution**: TBD

**Program 4:**

Given a list of integers, write a program using linq to find:

**Hint**: use aggregation operators like min, max, count, sum, average,

1. Count of odd numbers and evet numbers in the list
2. Count of unique numbers in the specified list
3. The Sum of all the specified numbers
4. The lowest and the greatest number in the specified list
5. The numbers that are greater than the average of all numbers in the specified list

**Sample Input:**

372, 409, 418, 282, 425, 496, 201, 161, 237, 149, 216, 296, 259, 128, 311, 99, 53, 183, 64, 15, 158, 491, 188, 94, 220, 394, 356, 352, 329, 398, 366, 327, 377, 342, 59, 116, 30, 44, 403, 366, 97, 431, 57, 396, 373, 54, 74, 258, 7, 155, 186, 184, 358, 134, 275, 381, 221, 325, 63, 31, 140, 348, 331, 338, 282, 389, 419, 417, 69, 195, 69, 10, 280, 198, 417, 226, 434, 292, 221, 166, 320, 99, 345, 174, 213, 183, 468, 243, 67, 178, 395, 205, 480, 372, 226, 248, 101, 219, 114, 483, 135, 157, 190, 312, 87, 176, 403, 161, 217, 49, 446, 428, 105, 396, 210, 42, 3, 157, 126, 100, 421, 177, 351, 390, 300, 363, 29, 408, 471, 169, 126, 216, 179, 237, 32, 164, 408, 34, 88, 37, 280, 308, 27, 18, 232, 466, 101, 85, 162, 185, 246, 443, 93, 15, 63, 458, 306, 299, 12, 223, 147, 494, 189, 48, 423, 230, 250, 367, 364, 47, 133, 469, 69, 90, 319, 113, 150, 333, 159, 333, 263, 444, 247, 286, 423, 183, 123, 164, 175, 74, 358, 110, 22, 365, 389, 285, 120, 479, 95, 3, 217, 233, 220, 116, 53, 276, 374, 4, 195, 450, 308, 472, 140, 179, 459, 203, 248, 403, 124, 241, 107, 86, 193, 291, 188, 23, 447, 204, 201, 35, 493, 171, 238, 156, 116, 30, 224, 131, 233, 147, 308, 120, 469, 224, 265, 385, 469, 342, 405, 186, 37, 216, 355, 354, 395, 132, 41, 428, 427, 31, 329, 170, 303, 161, 457, 137, 235, 484, 64, 430, 478, 118, 170, 476, 432, 487, 116, 468, 26, 303, 53, 105, 299, 15, 497, 277, 316, 384, 261, 483, 489, 295, 204, 203, 318, 138, 316, 308, 267, 128

**Sample Output:** To be printed in the console

Odd Number count: XXX

Even Number count: XXX

Unique numbers count: XXX

Sum: XXXXXXXX

Lowest Number: X

Greatest Number: XXX

Numbers greater than average of the list: XX, XX, XX, XX etc.

**Sample Solution:** TBD

**Program 5:**

Given 2 list of words write a program to find:

1. the words present in first dictionary and not in the second dictionary
2. the words present in the second dictionary and not first dictionary
3. common words in both the dictionary

2 sample files are provided below from where the input for the program could be hard coded as list of words.

**Hint:** Use Set Operations

**Sample Input**:



**Sample Output**: To be printed on console

Words present in first list and not in the second list:

*<List of words to go here>*

Words present in second list and not in the first list:

*<List of words to go here>*

Common words in both the dictionary

*<List of words to go here>*

**Sample Solution:** TBD