1. Sum Common Elements

Write a program to read two int arrays, eg. A{2,3,5,1} and B{1,3,9}, and to find out sum of common elements in given arrays. Print the sum, or print “No common elements found” if there are no common elements.
Assume the common element appears only once in each array.

Include a class UserProgramCode with a static method getSumOfIntersection which accept the size of two integer arrays and the two integer arrays. The return type (integer) should return the sum, or -1, accordingly.

Create a Class Program which would be used to accept two integer arrays, and call the static method present in UserProgramCode.

Input and Output Format:
Input consists of n+m+2 integers, where first two integers corresponds to the size of the two array lists, respectively, followed by the corresponding array elements.
Output consists of an Integer(the corresponding output) or string - (“No common elements found”).

Refer sample output for formatting specifications.

Sample Input 1:
4
3
2
3
5
1
1
3
9
Sample Output 1:
4

Sample Input 2:
4
3
2
31
5
14
1
3
9
Sample Output 2:
No common elements found

Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace Fwd_Prgs
{
    public class UserProgramCode
    {
        public static int getSumOfIntersection(int n1, int n2, int[] a, int[] b)
        {
            int sum = 0;
            for (int i = 0; i < n1; i++)
            {
                for (int j = 0; j < n2; j++)
                {
                    if (a[i] == b[j])
                    {
                        sum = sum + a[i];
                    }
                }
            }
            if (sum == 0)
            {
                return -1;
            }
            else
            {
                return sum;
            }
        }
    }


    class Program
    {
        static void Main(string[] args)
        {
            int n1 = int.Parse(Console.ReadLine());
            int n2 = int.Parse(Console.ReadLine());
            int[] a = new int[n1];
            int[] b = new int[n2];
            for (int i = 0; i < n1; i++)
            {
                a[i] = int.Parse(Console.ReadLine());
            }
            for (int i = 0; i < n2; i++)
            {
                b[i] = int.Parse(Console.ReadLine());
            }
            int res = UserProgramCode.getSumOfIntersection(n1, n2, a, b);
            if (res == -1)
            {
                Console.WriteLine("No common elements found");
            }
            else
            {
                Console.WriteLine(res);
            }
        }
    }
}
```

2. Add non Common Elements

Write a program to read two integer arrays and to add all the non common elements from the 2 integer arrays. Print the final output.

Example:
input1: [7,9,1,0]
input2: [10,6,5]
Output1:38
Business Rules:
Only positive numbers should be given to the input Lists.
1. If the input1 List consists of negative numbers, return -1.
2. If the input2 List consists of negative numbers, return -2.
3. If the both the input lists consists of negative numbers, return -3.

Include a class UserProgramCode with a static method which accepts the inputs in the following order (input1, size1, input2, size2). The return type (integer) should return output according to the business rules.

Create a Class Program which would be used to accept two lists, and call the static method present in UserProgramCode.

Input and Output Format:
Input consists of n+m+2 integers, where first two integers corresponds to the size of the two array lists, respectively, followed by the corresponding array elements.
Output consists of an Integer(the corresponding output), or a String "Input 1 has negative numbers" if the first array contains negative numbers, "Input 2 has negative numbers" if the second array contains negative numbers, or "Both inputs has negative numbers" if both array has negative numbers.
Refer sample output for formatting specifications.

Sample Input 1:
4
3
6
9
2
1
10
7
5
Sample Output 1:
38

Sample Input 2:
4
3
-6
9
2
1
10
7
5

Sample Output 2:
Input 1 has negative numbers

Sample Input 3:
4
3
6
9
2
1
10
-7
5
Sample Output 3:
Input 2 has negative numbers

Sample Input 4:
4
3
6
9
-2
1
10
-7
5
Sample Output 4:
Both inputs has negative numbers

Code :

```
using System;
using System.Collections.Generic; using Syst
em.Linq;
using System.Text;

namespace non_common_
{
 public class UserProgramCode
   {
        public static int sumNonCommonElement(int[] arr1, int size1, int[] arr2, int size2)
        {
            int sum = 0, a = 0, b = 0;
            for (int i = 0; i < size1; i++)
            {
                if (arr1[i] < 0)
                {
                    a = 0;
                    break;
                }
                else
                {
                    a = 1;
                }
            }
            for (int j = 0; j < size2; j++)
            {
                if (arr2[j] < 0)
                {
                    b = 0;
                    break;
                }
                else
                {
                    b = 1;
                }
            }
            if (a == 1 && b == 1)
            {
                int[] op = arr1.Except(arr2).Union(arr2.Except(arr1)).ToArray();
                foreach (int item in op)
                {
                    sum = sum + item;
                }
                return sum;
            }
            if (a == 0 && b == 1) return -1;
            else if (a == 1 && b == 0) return -2;
            if (a == 0 && b == 0) return -3;
            return 0;
        }

    }


    class Program
    {
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());
            int m = int.Parse(Console.ReadLine());
            int[] a1 = new int[n];
            int[] a2 = new int[m];
            for (int i = 0; i < n; i++)
            {
                a1[i] = int.Parse(Console.ReadLine());
            }
            for (int i = 0; i < m; i++)
            {
                a2[i] = int.Parse(Console.ReadLine());
            }

            int flag = UserProgramCode.sumNonCommonElement(a1, n, a2, m);
            if (flag == -1)
            {
                Console.WriteLine("Input 1 has negative numbers");
            }
            else if (flag == -2)
            {
                Console.WriteLine("Input 2 has negative numbers");
            }
            else if (flag == -3)
            {
                Console.WriteLine("Both inputs has negative numbers");
            }
            else
            {
                Console.WriteLine(flag);
            }
        }
     }
```

3. Bonus Calculation

Write a program to calculate the bonus of the employee given the basic salary of the employee.
The bonus will be calculated based on the below category.

If Basic Salary>15000 and less than 20001 calculate bonus as 17% of basic+1500 If Basic Salary>10000 and less than 15001 calculate bonus as 15% of basic+1200 If Basic Salary<10001 calculate bonus as 12% of basic+1000, for rest calculate bonus as 8%of basic+500

Business rule:

1. If the salary given is a negative number, then print -1.
2. If the salary given is more than 1000000, then print -2 .
3. All the test cases has the calculated bonus as integer value only.

Create a class named UserProgramCode that has the following static method public static int calculateBonus(int input1)
Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:
Input consists of an integer that corresponds to the salary. Output is an integer.
Refer sample output and business rule for output formatting specifications.

Sample Input 1 :
10000
Sample Output1 :
2200

Sample Input 2 :
2000000
Sample Output 2 :
-2

Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace calculate_Bonus
{
    public class UserProgramCode
    {
        public static int calculateBonus(int input1)
        {
            int bonus;
            if (input1 > 15000 && input1 < 20001)
            {
                bonus = Convert.ToInt32((0.17 * input1) + 1500);
            }
            else if (input1 > 10000 && input1 < 15001)
            {
                bonus = Convert.ToInt32((0.15 * input1) + 1200);
            }
            else if (input1 < 10001)
            {
                bonus = Convert.ToInt32((0.12 * input1) + 1000);
            }
            else
            {
                bonus = Convert.ToInt32((0.08 * input1) + 500);
            }
            return bonus;
        }
    }


    class Program
    {
        static void Main(string[] args)
        {
            int sal = Convert.ToInt32(Console.ReadLine());
            if (sal < 0)
            {
                Console.WriteLine(-1);
            }
            else if (sal > 1000000)
            {
                Console.WriteLine(-2);
            }
            else
            {
                int res = UserProgramCode.calculateBonus(sal);
                Console.WriteLine(res);
            }
        }
    }
}
```

4. Sort the list

Write a program which reads an Integer(size of the list), a String List and a character, and to get the strings that will not start with the given character irrespective of case. Sort the elements in ascending order based on its length. Print the output list. If the elements are having the same length, then display the elements in alphabetical order.
Include a class **UserProgramCode** with static method **GetTheElements** which accepts a String list and a character. The return type is List<String>.
Create a Class Program which would be used to get the inputs and call the static method present in UserProgramCode. In **GetTheElements** method
Only alphabets should be given in list , otherwise return "Invalid Input". When the output list is empty, then return "List is Empty". Otherwise return the appropriate result.
In **Program** class Print the result which is return by **GetTheElements** method in **UserProgramCode.**

Input output format
The first line of the input is an integer that corresponds to n, the size of the list. The next n lines of input correspond to the elements in the string list. The line of the input contains the character. The output is the List type List<String>.

Sample Input 1: 3 read write edit e
Sample Output 1: read write

Sample Input2 : 2 Elegent event e
Sample Output2 : List is Empty

Sample Input 3: 2 Eleg$ent e^ent e
Sample Output 3 : Invalid Input

Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
namespace l2ndl3
{
    public class UserProgramCode
    {
        public static List<String> GetTheElements(List<String> li, char c)
        {
            List<string> res = new List<string>();
            bool samelength = false;
            int len1 = li[0].Length;
            foreach (string str in li)
            {
                if (str.Length == len1)
                {
                    samelength = true;
                }
                else
                {
                    samelength = false;
                    break;
                }
            }

            if (samelength)
            {
                res = (from z in li
                       where !z.StartsWith(c.ToString().ToUpper()) &&
!z.StartsWith(c.ToString().ToLower())
                       select z).ToList();
                res.Sort();
            }
            else
            {
                res = (from z in li
                       where !z.StartsWith(c.ToString().ToUpper()) &&
!z.StartsWith(c.ToString().ToLower())
                       orderby z.Length
                       select z).ToList();
            }
            return res;
        }
    }


    class Program
    {
        static void Main(string[] args)
        {
            List<string> li = new List<string>();
            int n = int.Parse(Console.ReadLine());
            for (int i = 0; i < n; i++)
            {
                li.Add(Console.ReadLine());
            }
            char c = char.Parse(Console.ReadLine());

            bool isalpha = false;
            foreach (string st in li)
            {
                if (st.All(char.IsLetter))
                {
                    isalpha = true;
                }
                else
                {
                    isalpha = false;
                    break;
                }
            }

            if (isalpha)
            {
                List<string> li2 = UserProgramCode.GetTheElements(li, c);
                if (li2.Any())
                {
                    foreach (var item in li2)
                    {
                        Console.WriteLine(item);
                    }
                }
                else
                {
                    Console.WriteLine("List is Empty");
                }
            }
            else
            {
                Console.WriteLine("Invalid Input");
            }
        }
    }
}
```

5. Image Types

Given a string array input which consists of image file names along with their respective image type extensions in the format ("filename.extensiontype",..so on). The image file name and the extension are seperated by a dot (.)operator. Write a program to calculate the count of image files having same extension type and store the values in the output string array variable in the below format.
output (Key,Value) = (ExtensionType1,count1,ExtensionType2,count2,...so on) . Output should be stored in descending order based on the count of image files having the same extension type.

Note: jpeg,jfif,exif,tiff,raw,gif,bmp,png are the various types of image file extensions

Business Rules:

1)If all the elements of the input array do not have image type extension, then print -1. 2)If any of the file name doesn't contain extension type or if the extension is not an image type then it will be treated as other type, take the count of all such files and store as the last element in the sorted output array with key element as "others" and value element as the calculated count.
3)If more than one key element have same count, then store the key and their respective value element in the order given in input.

Create a class named UserProgramCode that has the following static method public static List<string> countImageTypes(string[] input1)
Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:
The first line of the input consists of an integer, n that corresponds to the number of elements in the input array .
The next 'n' lines of input correspond to elements in the input array.
Refer business rules and sample output for formatting specifications.

Sample Input 1 :

4
Employee.jpeg Purchase.jpeg
stock.jpeg book.gif

Sample Output 1 :

jpeg 3
gif 1

Sample Input 2 :

7
Sales.doc Employee.jpeg
Purchase.jpeg image.png
stock.jpeg book.gif
pen

Sample Output 2 :

jpeg 3
png 1
gif 1
others 2

Code :

```
using System;
using System.Collections.Generic;

namespace imageTypes
{
    class Program
    {
        public class UserProgramCode
        {
            public static List<string> countImageTypes(string[] st, int n)
            {
                List<string> l = new List<string>();
                List<string> l1 = new List<string>();
                int count = 0, c = 0;
                List<string> st2 = new List<string>
{ "jpeg", "jfif", "exif", "tiff", "raw", "bmp", "png" , "gif" };
                string[] st1 = new string[2];
                foreach (var item in st)
                {
                    if (item.Contains('.'))
                    {
                        st1 = item.Split('.');
                        for (int i = 0; i < 2; i++)
                        {
                            l.Add(st1[i]);
                        }
                    }
                    else
                    {
                        l.Add(item);
                        l.Add("other");
                    }

                }
                for (int i = 0; i < 8; i++)
                {
                    count = 0;
                    c = 0;
                    for (int j = 1; j <= l.Count; j += 2)
                    {
                        if (st2[i] == l[j])
                        {
                            count++;
                        }
                        else if (st2[i] != l[j] && !(st2.Contains(l[j])))
                        {
                            c++;
                        }
                    }
                    if (count != 0)
                    {
                        l1.Add(st2[i]);
                        l1.Add(count.ToString());
                    }
                }
                if (c != 0)
                {
                    l1.Add("others");
                    l1.Add(c.ToString());
                }
                return l1;
            }
        }
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());
            string[] input1 = new string[n];
            for (int i = 0; i < n; i++)
            {
                input1[i] = Console.ReadLine();
            }
            List<string> countimagetypes = UserProgramCode.countImageTypes(input1, n);
            foreach (var item in countimagetypes)
            {
                Console.WriteLine(item);
            }
        }
    }
}
```

6. Find N’th Largest Number –

```
using System;

namespace Nth_Largest
{
    class userProgramcode
    {
        public static int findNthLargestNumber(int[] input1, int input2)
        {
            foreach (int item in input1)
            {
                if (item < 0)
                {
                    return -1;
                }
            }
            if (input2 < 0)
            {
                return -1;
            }
            Array.Sort(input1);
            Array.Reverse(input1);
            return input1[input2 - 1];
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            int size = int.Parse(Console.ReadLine());
            int[] arr = new int[size];
            for (int i = 0; i < size; i++)
            {
                arr[i] = int.Parse(Console.ReadLine());
            }
            int n = int.Parse(Console.ReadLine());
            int nthlargest = userProgramcode.findNthLargestNumber(arr, n);


         if (nthlargest == -1)
            {
                Console.WriteLine("Invalid Input");
            }
            else
            {
                Console.WriteLine(nthlargest);
            }
        }
    }
}
```