1)Add Non Common Element:

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

namespace ConsoleApplication32
{

    public class UserProgramCode
    {
     public static int sumNonCommonElement(int[] ar1, int n, int[] ar2, int m)
     {
         int a = 0, b = 0;
         int[] temp = new int[m + n];
         for (int i = 0; i < n; i++)
         {
             if (ar1[i] < 0)
                 a = 1;

         }
         for (int j = 0; j < m; j++)
         {
             if (ar2[j] < 0)
                 b = 1;

         }
         if (a == 1 && b == 0)
         {
             return -1;
         }
         else if (a == 0 && b == 1)
         {
             return -2;
         }
         else if (a == 1 && b == 1)
         {
             return -3;
         }
         else if (a == 0 && b == 0)
         {
             for (int i = 0; i < n; i++)

                 for (int j = 0; j < m; j++)

                     if (ar1[i] == ar2[j])
                     {
                         ar1[i] = 0;
                         ar2[j] = 0;
                     }
         }
         return ar1.Sum() + ar2.Sum();
     }
    }

    class NonCommonElement
```

```csharp
    {
        static void Main(string[] args)
        {

            int n = int.Parse(Console.ReadLine());
            int m = int.Parse(Console.ReadLine());
            int[] ar1 = new int[n];
            int[] ar2 = new int[m];
            for (int i = 0; i < n; i++)
                ar1[i] = int.Parse(Console.ReadLine());
            for (int i = 0; i < m; i++)
                ar2[i] = int.Parse(Console.ReadLine());
            int flag = UserProgramCode.sumNonCommonElement(ar1, n, ar2, 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);
            Console.ReadLine();
        }
    }
}
```

2)Bonus Calculation.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace bonus

{

```csharp
    public class UserProgramCode
    {
     public static int CalculateBonus(int basic)
     {
         int bonus = 0;
         if (basic < 0)
             return -1;
         if (basic > 1000000)
             return -2;
         if (basic < 20001 && basic > 15000)
         {
             bonus = Convert.ToInt32(basic * 0.17) + 1500;
         }
         if (basic > 10000 && basic < 15001)
             bonus = Convert.ToInt32(basic * 0.15) + 1200;
```

```csharp
            if (basic < 10001)
                bonus = Convert.ToInt32(basic * 0.12) + 1000;
            else

                bonus = Convert.ToInt32(basic * 0.08) + 500;
            return bonus;
        }
    }
}
```

calculate bonus

```csharp
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace bonus

{


    class Program
    {
        static void Main(string[] args)
        {
            int basic = int.Parse(Console.ReadLine());
            int op = UserProgramCode.CalculateBonus(basic);
            Console.WriteLine(op);
            Console.ReadLine();
        }
    }
}
```

3)Find nth Largest Number:

```csharp
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace TestPractice

{

public class UserProgramCode
    {
        public static int output1;
        public static int nthLargest(int[] input1, int input2)
        {
            if (input2 < 1)
```

```csharp
            {
                return -1;
            }
            foreach (var item in input1)
            {
                if (item < 0)
                {
                    output1 = -1;
                    return output1;
                }
            }
            if (output1 != -1)
            {
                Array.Sort(input1);
                Array.Reverse(input1);
                input1 = input1.Distinct().ToArray();
                output1 = input1[input2 - 1];
            }
            return output1;
        }
    }

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;


namespace TestPractice

{

class Program1
    {
        static void Main(string[] args)
        {
            int n;
            n = Convert.ToInt32(Console.ReadLine());
            int[] input1 = new int[n];
            for (int i = 0; i < n; i++)
            {
                input1[i] = Convert.ToInt32(Console.ReadLine());
            }
            int input2 = Convert.ToInt32(Console.ReadLine());
            int res = UserProgramCode.nthLargest(input1, input2);
            if (res == -1)
            {
                Console.WriteLine("invalid input");
            }
            else
            {
                Console.WriteLine(res);
            }
            Console.ReadLine();
        }
```

```
        }
}
```

4)Sum Common Element:

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;
        }

    }
}
```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Fwd_Prgs

{

```
class Program3
    {
        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);
            Console.ReadLine();
        }
    }
}
```

5)Sort the list:

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


namespace ConsoleApplication22
{
    class UserProgramCode
    {
        public static List<string> GetTheElements(string[] t, char f)
        {
            Regex r = new Regex(@"^([a-zA_Z]{1,})$");
            List<string> m = new List<string>();
            foreach (string c in t)
            {
                if (!r.IsMatch(c))
                {
                    m.Add("-1");
                    break;
                }
                else
                {
                    if (!c.StartsWith(f.ToString()))
                    {
                        m.Add(c);
                    }
                }
            }
            for (int i = 0; i < m.Count; i++)
            {
                for (int j = i + 1; j < m.Count; j++)
                {
```

```csharp
                    if (m[i].Length == m[j].Length)
                    {
                        m.Sort();
                    }
                    else
                    {
                        if (m[j].Length < m[i].Length)
                        {
                            m.Reverse();
                        }
                    }
                }
            }
            return m;
        }
    }

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

namespace ConsoleApplication22
{
class Program2
    {
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());
            string[] p = new string[n];
            for (int i = 0; i < n; i++)
            {
                p[i] = Console.ReadLine();
            }
            char e = char.Parse(Console.ReadLine());
            List<string> l = new List<string>();
            l = UserProgramCode.GetTheElements(p, e);
            if (l.Contains("-1"))
            {
                Console.WriteLine("Invalid Input");
            }
            else if (l.Count > 0)
            {
                foreach (string x in l)
                {
                    Console.WriteLine(x);
                }
            }
            else
                Console.WriteLine("list is empty");
            Console.WriteLine();
        }
    }
}
```

6)Image Types:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace image

{

  class userProgramcode

    {

```csharp
        public static List<string> imagescount(List<string> input1)
        {
            int k = 0, ctr = 0;
            string[] ar = new string[8] { "jpeg", "jfif", "exif", "tiff", "raw",
"gif", "bmp", "png" };
            List<string> outp = new List<string>();
            string[] st = new string[input1.Count];
            int[] sco = new int[input1.Count];
            for (int j = 0; j < input1.Count; j++)
            {
                string[] arr = input1[j].Split('.');
                if (arr.Length == 2 && !st.Contains(arr[1]) && ar.Contains(arr[1]))
                {

                    st[k] = arr[1];
                    sco[k] = sco[k] + 1;
                    k++;
                }
                else if (arr.Length == 2 && st.Contains(arr[1]) &&
ar.Contains(arr[1]))
                {
                    for (int p = 0; p < st.Length; p++)
                    {
                        if (st[p] == arr[1])
                        {
                            sco[p] = sco[p] + 1;
                            break;
                        }
                    }
                }
                else
                {
                    ctr++;
```

```csharp
                }
            }

            if (ctr != input1.Count)
            {
                int[] co = new int[k];
                co = sco.ToArray();
                sco = co.Distinct().ToArray();
                Array.Sort(sco);
                Array.Reverse(sco);

                for (int m = 0; m < sco.Length - 1; m++)
                {
                    for (int n = 0; n < co.Length; n++)
                    {
                        if (sco[m] == co[n])
                        {
                            outp.Add(st[n]);
                            outp.Add(sco[m].ToString());
                        }
                    }
                }
            }
            if (ctr != 0 && ctr != input1.Count)
            {
                outp.Add("Others");
                outp.Add(ctr.ToString());
            }
            else if (ctr == input1.Count)
            {
                outp.Add("-1");
                return outp;
            }

            return outp;
        }
    }

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;



namespace image

{

class Program4
```

```csharp
    {
        static void Main(string[] args)
        {
            int a = int.Parse(Console.ReadLine());
            List<string> image = new List<string>();
            for (int i = 0; i < a; i++)
            {
                image.Add(Console.ReadLine());
            }
            List<string> ouyp = userProgramcode.imagescount(image);

            foreach (string s in ouyp)
            {

                Console.WriteLine(s);
            }
            Console.ReadLine();

        }

    }

}
```