

```

using
System;

namespace Assignment_1___Basics
{
    class Program
    {
        static void Main(string[] args)
        {
            // Exercise 1 : Computing Prime Numbers and handling Exceptions.
            // summary      : This method prints all the prime numbers between x
and y
            // For example 5, 25 will print all the prime numbers between 5 and
25 i.e.
            // 5, 7, 11, 13, 17, 19, 23
            // Tip: Write a method isPrime() to compute if a number is prime or
not.

            Console.WriteLine("Hello World!");
            int start, end;
            Console.Write("Enter lower range: ");
            start = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter upper range: ");
            end = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Prime numbers between {0} and {1} are: ", start,
end);

            Console.WriteLine("=====");
            printPrime(start, end);

            Console.WriteLine("\n=====");

            // Exercise 2 : Print the Sum of the series.
            // para      n - number of terms of the series, integer (int)
            // summary      : This method computes the series  $1/2 - 2!/3 + 3!/4 - 4!/5$  --- n
            // where ! means factorial, i.e.,  $4! = 4*3*2*1 = 24$ . Round off
the results to
            // three decimal places.
            Console.Write("\nEnter the number n for which you want to see the
series (Enter Number Greater than 0): ");
            int numn;
            numn = Convert.ToInt32(Console.ReadLine());
            getSeriesResult(numn);

```

```

Console.WriteLine("\n=====");

    // Exercise 3 : Decimal to Binary Conversion
    Console.Write("\nEnter the Decimal number which you want to convert
to Binary : ");
    int bin;
    bin = Convert.ToInt32(Console.ReadLine());
    decimalToBinary(bin);

Console.WriteLine("\n=====");

    // Exercise 4 : Decimal to Binary Conversion
    Console.Write("\nEnter the Decimal number which you want to convert
to Binary : ");
    int dec;
    dec = Convert.ToInt32(Console.ReadLine());
    binarytoDecimal(dec);

Console.WriteLine("\n=====");

    // Exercise 5 : Triangle Print with n as input
    Console.Write("Enter the Range for the Traingle you want to print
=");

    int n = int.Parse(Console.ReadLine());
    printTriangle(n);

Console.WriteLine("\n=====");

    // Exercise 6 : Compute the frequency of an element in an array.
    Console.WriteLine("How many numbers do you wish to enter");
    string param = Console.ReadLine();
    int userInput = Convert.ToInt32(param);
    try
    {
        if (userInput <= 0)
        {
            throw (new ZeroNumException("Zero arguments entered"));
        }
        else
        {
            int[] a = new int[userInput];
            Console.Write("\n Input elements in the array :\n");

```

```

        for (int i = 0; i < userInput; i++)
        {
            Console.Write("element - {0} : ", i);
            a[i] = Convert.ToInt32(Console.ReadLine());
        }
        computeFrequency(a);
    }

    catch (ZeroNumException)
    {
        Console.Write("You have entered Zero Arguments. Please re-enter
the number again.");

    }

    Console.ReadKey();
}

static bool isPrime(int n)
{
    // Corner case
    if (n <= 1)
        return false;

    // Check from 2 to n-1
    for (int i = 2; i < n; i++)
        if (n % i == 0)
            return false;

    return true;
}

static void printPrime(int a,int b)
{
    try
    {
        if(a>b)
        {
            throw (new LargeNumException("Improper start and end number
input"));

```

```

    }
    else
    {
        for (int i = a; i <= b; i++)
        {
            if (isPrime(i))
                Console.Write(i + " ");
        }
    }

}

catch(LargeNumException)
{
    Console.WriteLine("End > Start Number ! Press any key to quit!");
}

}

public static void getSeriesResult(int n)
{
    double temp = 0;
    try
    {
        if (n==0)
        {
            throw (new ZeroNumException(" No Valid Input"));
        }
        else
        {
            for(int i=1;i<=n;i++)
            {
                double ch;
                ch = factorial(i);

                ch = ch / (i + 1);

                if ( i % 2 == 0) // Checking Even
                {
                    temp = temp - ch;
                }
                else // Checking Odd

```

```

        {
            temp = temp + ch;
        }
    }
    Console.WriteLine("\nThe sum of the series for the {0} ", n +
" is : "+ temp);
    }

    }
    catch(ZeroNumException)
    {
        Console.WriteLine("You entered 0 as the input. So the result is
0");
    }

}

static int factorial(int n)
{
    if (n == 0)
        return 1;

    return n * factorial(n - 1);
}

// This function converts Binary to Decimal.
public static long decimalToBinary(long n)
{
    long remainder;
    string result = string.Empty;
    while (n > 0)
    {
        remainder = n % 2;
        n /= 2;
        result = remainder.ToString() + result;
    }
    Console.WriteLine("Binary: {0}", result);
    return 0;
}

public static long binarytoDecimal(long num)
{

```

```

    long binVal, decVal = 0, baseVal = 1, rem;

    binVal = num;
    while (num > 0)
    {
        rem = num % 10;
        decVal = decVal + rem * baseVal;
        num = num / 10;

        baseVal = baseVal * 2;
    }
    Console.Write("Binary Number: " + binVal);
    Console.WriteLine("\nDecimal: " + decVal);
    Console.ReadLine();
    return 0;
}

public static void printTriangle(int n)
{
    int i, j, k, l;

    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n - i; j++)
        {
            Console.Write(" ");
        }
        for (k = 1; k <= i; k++)
        {
            Console.Write("*");
        }
        for (l = i - 1; l >= 1; l--)
        {
            Console.Write("*");
        }
        Console.WriteLine("\n");
    }
}

public static void computeFrequency(int[] arr)
{
    int i=0 , n = arr.Length;

```

```

while (i < n)
{
    // If this element is already
    // processed, then nothing to do
    if (arr[i] <= 0)
    {
        i++;
        continue;
    }

    // Find index corresponding to
    // this element. For example,
    // index for 5 is 4
    int elementIndex = arr[i] - 1;

    // If the elementIndex has an element
    // that is not processed yet, then
    // first store that element to arr[i]
    // so that we don't lose anything.
    if (arr[elementIndex] > 0)
    {
        arr[i] = arr[elementIndex];

        // After storing arr[elementIndex],
        // change it to store initial count
        // of 'arr[i]'
        arr[elementIndex] = -1;
    }
    else
    {
        // If this is NOT first occurrence
        // of arr[i], then increment its count.
        arr[elementIndex]--;

        // And initialize arr[i] as 0 means
        // the element 'i+1' is not seen so far
        arr[i] = 0;
        i++;
    }
}

Console.WriteLine("\nBelow are counts of " +
    "all elements" + "\n");

```

```

        for (int j = 0; j < n; j++)
            Console.Write(j + 1 + " Occurs " + Math.Abs(arr[j]) + "
times\n");

    }

}

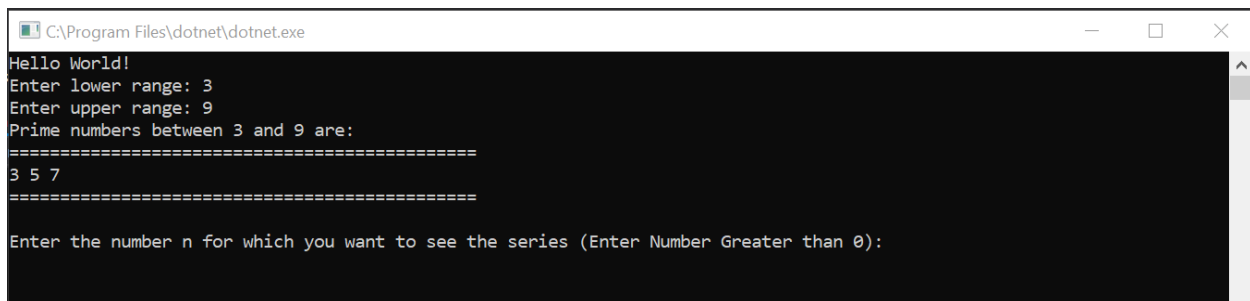
}

// Exception Handling for the first two cases
public class ZeroNumException : Exception
{
    public ZeroNumException(string message) : base(message)
    {
    }
}

public class LargeNumException : Exception
{
    public LargeNumException(string message) : base(message)
    {
    }
}

```

OUTPUT:



```

C:\Program Files\dotnet\dotnet.exe
Hello World!
Enter lower range: 3
Enter upper range: 9
Prime numbers between 3 and 9 are:
=====
3 5 7
=====
Enter the number n for which you want to see the series (Enter Number Greater than 0):

```


C:\Program Files\dotnet\dotnet.exe

=====

How many numbers do you wish to enter

5

Input elements in the array :

element - 0 : 1

element - 1 : 1

element - 2 : 1

element - 3 : 2

element - 4 : 3

Below are counts of all elements

1 Occurs 3 times

2 Occurs 1 times

3 Occurs 1 times

4 Occurs 0 times

5 Occurs 0 times