First Question:

using System;

using System.Collections.Generic;

namespace Assignment1\_S19

{

class Program

{

public static void Main()

{

Console.WriteLine("please Enter the Starting Number"); //getting the starting number from user

string StartingNumber=Console.ReadLine();

Console.WriteLine("please Enter the Ending Number"); //getting the ending number from user

string EndingNumber = Console.ReadLine();

double Starting = Convert.ToDouble(StartingNumber);

double Ending = Convert.ToDouble(EndingNumber);

List <int> PrimeList = new List <int>(); // creating a list to hold all prime numbers in the interval

int IntegerEnding = Convert.ToInt32(Ending);

int IntegerStarting= Convert.ToInt32(Starting) ;

int Counter = IntegerStarting;

if (Starting<Ending) //to check if the interval is valid

{

if(Ending - (IntegerEnding = Convert.ToInt32(Math.Floor(Ending)))==0 && Starting - (IntegerStarting = Convert.ToInt32(Math.Floor(Starting))) == 0) //to check if the entered values are integers

{

while (Counter <= IntegerEnding)

{

bool PrimeNumber= IsPrime(Counter); // calling the IsPrime method

if (PrimeNumber==true)

{

PrimeList.Add(Counter); // adding prime numbers to the prime list

Counter += 1;

}

else

{

Counter += 1;

}

}

foreach(int t in PrimeList) //printing prime numbers in the list

{

Console.WriteLine(t);

}

Console.ReadLine();

}

else

{

Console.WriteLine("Numbers Must be Integer"); //inform the user that entered values are not integer

Console.ReadLine();

}

}

else

{

Console.WriteLine("Ending Number Must be Greater than Starting Number"); //Inform the user that the interval is not valid

Console.ReadLine();

}

}

static private bool IsPrime( int TheNumber) // defining the method to check for the passed argument to be prime

{

if (TheNumber==1)

{

return true;

}

else if (TheNumber==2)

{

return true;

}

else

{

for (int i =2; i<TheNumber; i++)

{

int aa = TheNumber;

aa %= i;

if (aa==0)

{

return false;

}

else

{

aa = TheNumber;

}

}

return true;

}

}

}

}

Second Problem:

using System;

namespace first\_assignment\_second\_question

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("please Enter the Desired number"); // reading the series number from user

int Number = Convert.ToInt32( Console.ReadLine());

double ComputedSeries = 0; // creating the series sum

for (int i=1; i<(Number+1); i++)

{

ComputedSeries = ComputedSeries+ ((Convert.ToDouble (Math.Pow(-1, i - 1))) \* ((Convert.ToDouble ((Factorial(i))) / (Convert.ToDouble (i + 1))))); //calling the defined factorial method to compute the series

}

Console.WriteLine("the result is" +" "+ ComputedSeries);

Console.ReadLine();

}

static private int Factorial( int num) // defining the factorial calculator method

{

if (num==1)

{

return 1;

}

else

{

return Factorial(num - 1) \* num;

}

}

}

}

Third Problem

using System;

using System.Collections.Generic;

namespace first\_assignment\_third\_question

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please Enter Your Number"); //asking the user for the desired number to be converted

string Number= Console.ReadLine();

int IntNumber = Convert.ToInt32(Number);

List<int> MyList = new List<int>(); // creating a list to hold the results of sequential divisions

int quotient= IntNumber;

bool condition = true;

while (condition==true)

{

if (quotient==0) //end of division

{

condition=false;

}

else if (quotient==1)

{

MyList.Add(1);

quotient = 0;

}

else

{

MyList.Add(quotient % 2);

quotient = quotient/2; //sequential division

}

}

MyList.Reverse(); // to put the binary results in the right order

var result = string.Join("", MyList); // to concatenate the binary results together to make a string of numbers

Console.WriteLine("The Binary Conversion of the Numer is");

Console.WriteLine(result);

Console.ReadLine();

}

}

}

Fourth Problem:

using System;

using System.Collections.Generic;

using System.Linq;

namespace first\_assignment\_fourth\_question

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please Enter a Binary Number");//getting numbers from users

string Number= Console.ReadLine();

List<string> SplittedNumber = new List<string>(Number.Select(c=>c.ToString()) ); //splitting the string into list to separate

int numberlength = SplittedNumber.Count ; // finding the length of the entered binary number

bool NumberValidity = true;

int Counter = 0;

int Result = 0;

for (int i = 0; i < numberlength; i++)

{

if (Convert.ToInt32( SplittedNumber[i]) > 1) //checking if the entered number was a binary or not

{

Console.WriteLine("The Entered Number is not Binary");

NumberValidity = false;

}

}

if(NumberValidity==true)

{

while(Counter<numberlength)

{

Result = Result+ ((Convert.ToInt32(SplittedNumber [numberlength-1- Counter])\* TwoToThePowerN(Counter))); //converting the binary number to decimal calling the defined method

Counter += 1;

}

}

Console.WriteLine(" The Converted Number Is" + "" + Result);

Console.ReadLine();

}

static private int TwoToThePowerN(int n)// Defined method to calculate 2^n

{

if(n==0)

{

return 1;

}

else

{

return 2 \* TwoToThePowerN(n - 1);

}

}

}

}

Fifth Question:

using System;

namespace First\_Assignment\_Fifth\_Question

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("How Many Rows Should be Printed?"); // reading the number of rows to be printed from user

string NumberOfRows=Console.ReadLine();

int IntRowNumber = Convert.ToInt32(NumberOfRows);

string starsrow = "";

string blankspaces = "";

string stars = "";

for (int i = 1; i <= IntRowNumber; i++) //creating a list of blank spaces and stars

{

stars = new string('\*', (2 \* i - 1));

blankspaces = new string(' ', (IntRowNumber - i));

starsrow = blankspaces + stars + blankspaces; // concatenating stars and blank spaces

Console.WriteLine(starsrow);

}

Console.ReadLine();

}

}

}

Six Question:

using System;

using System.Collections.Generic;

using System.Linq;

namespace First\_Assignment\_sixth\_Question

{

class Program

{

static void Main(string[] args)

{

string[] MyArray = { "A", "B", "C", "D", "E", "F", "G", "I", "J", "K","A" }; //Arbitrary List

List<String> ListOfArray = new List<string>(MyArray); //Converting the Array to a list

Dictionary<string, int> ReturnedDictionary = new Dictionary<string, int>(); // creating a dictionary to hold the objects along with their counts

ReturnedDictionary = CountOfObjects(ListOfArray); //calling the defined method of counting the repetition of objects

Console.WriteLine("Number"+" "+ "Frequency");

foreach (KeyValuePair<string, int> author in ReturnedDictionary)

{

Console.WriteLine(" {0}, {1}", author.Key, author.Value);

}

Console.ReadLine();

}

static private Dictionary<string,int> CountOfObjects (List<string> TheList)// list of the objects in the first array is passed to the method to count their repetition

{

Dictionary<string, int> DictionaryOfList = new Dictionary<string, int>();

int Count = TheList.Count;

for (int i=0; i<Count; i++) // checks if the key exists in the dictionary else adds to the dictionary and puts the value equal to the repetition count

{

if(DictionaryOfList.ContainsKey( TheList[i]))

{

DictionaryOfList[TheList[i]] += 1;

}

else

{

DictionaryOfList.Add(TheList[i], 1);

}

}

return DictionaryOfList;

}

}

}