1) Write a program that will take an input from user as number and print all the numbers

from 0 to the given number.

public void PrintNumbersFromZeroToGivenNumber() //Q1

{

int number = 0;

Console.WriteLine("Please type a number");

number = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("The numbers are:");

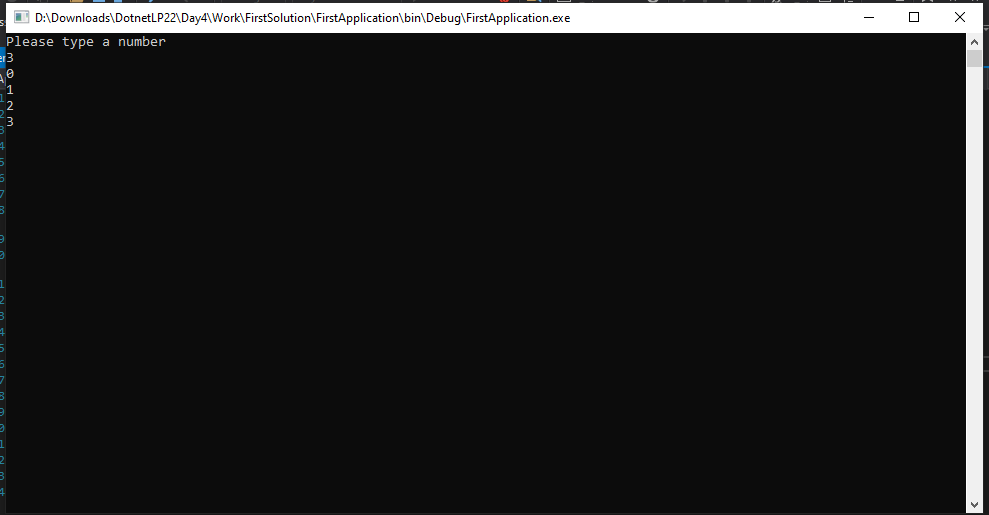
for (int i = 0; i <= number; i++)

{

Console.WriteLine(i);

}

}



2) Create a program that will find out if the given number is odd or even

public void CheckNumberEvenOrOdd() //Q2

{

int number = 0;

Console.WriteLine("Please type a number");

number = Convert.ToInt32(Console.ReadLine());

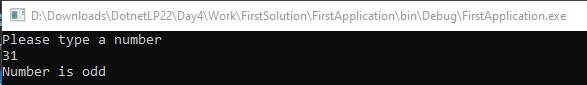
if (number % 2 == 1)

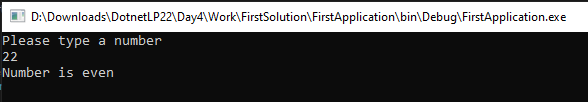
Console.WriteLine("Number is odd");

else

Console.WriteLine("Number is even");

}





3) Create a program that will take 2 numbers and find out the greatest of the 2

public void GetGreatestOfTwoNumbers()

{

int number1, number2;

Console.WriteLine("Please enter the first number");

number1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the second number");

number2 = Convert.ToInt32(Console.ReadLine());

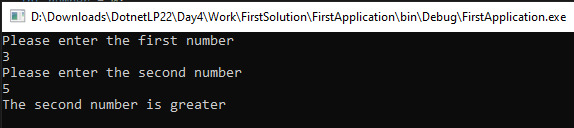
if(number1 > number2)

Console.WriteLine("The first number is greater");

else

Console.WriteLine("The second number is greater");

}



4) Improve the program written in question 3 to find the greatest of 3 numbers

public void GetGreatestOfThreeNumbers()

{

int number1, number2, number3, max = 0;

Console.WriteLine("Please enter the first number");

number1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the second number");

number2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the third number");

number3 = Convert.ToInt32(Console.ReadLine());

int[] arr = {number1, number2, number3};

for (int i = 0; i < 3; i++)

{

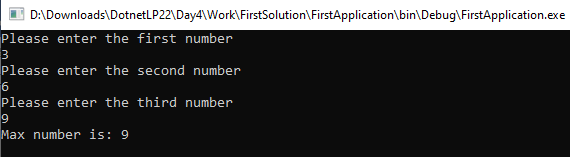
if (arr[i] > max)

max = arr[i];

}

Console.WriteLine("Max number is: " + max);

}



5) Take the minimum and maximum number from user and find all numbers inbetween

public void NumbersBetweenMinAndMaxNumber()

{

int min, max;

Console.WriteLine("Please enter the minimum number");

min = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the maximum number");

max = Convert.ToInt32(Console.ReadLine());

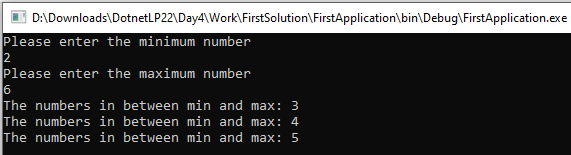
for (int i = min+1; i < max; i++)

{

Console.WriteLine("The numbers in between min and max: " + i);

}

}



6) Find if a given number is prime

public void FindPrimeNumberFromGivenNumber()

{

int number, flag = 0, i;

Console.WriteLine("Please enter a number");

number = Convert.ToInt32(Console.ReadLine());

for (i = 2; i <= number - 1; i++)

{

if (number % i == 0)

{

Console.WriteLine("Number is not a prime number");

flag = 1;

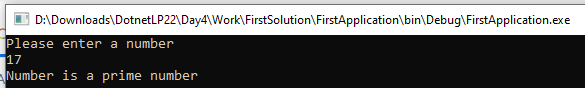
break;

}

}

if (i == number || flag == 0)

Console.WriteLine("Number is a prime number");

}

7) Improve the program in 5 to find all the prime numbers between the given numbers

public void NumbersBetweenMinAndMaxNumberWithPrimeDetection()

{

int min, max;

Boolean isPrime = false;

Console.WriteLine("Please enter the minimum number");

min = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the maximum number");

max = Convert.ToInt32(Console.ReadLine());

for (int i = min + 1; i < max; i++)

{

isPrime = FindPrimeNumberFromGivenNumberInput(i);

if (isPrime)

Console.WriteLine("The prime numbers in between min and max are: " + i);

}

}

public Boolean FindPrimeNumberFromGivenNumberInput(int primeNo)

{

int flag = 0, i;

Boolean isPrime = false;

for (i = 2; i <= primeNo - 1; i++)

{

if (primeNo % i == 0)

{

isPrime = false;

flag = 1;

break;

}

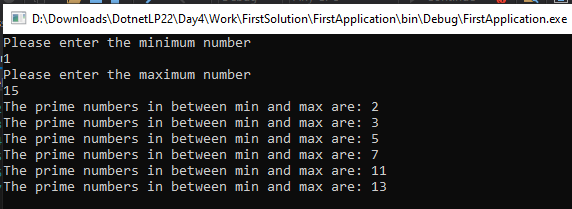
}

if (i == primeNo || primeNo == 0)

isPrime = true;

return isPrime;

}



8) Take input from user until the user enters a negative number and find the sum of all the numbers

that are divisible by 7

public void GetNegativeNumberAndCheckEachNumberDivisibleBy7()

{

int number = 1, sum = 0, count = 0;

List<int> list = new List<int>();

while (number > 0)

{

count++;

list.Add(number);

Console.WriteLine("Please enter a negative number");

number = Convert.ToInt32(Console.ReadLine());

}

list.RemoveAt(0); //remove first element as it was added inside

for (int j = 0; j < list.Count; j++)

{

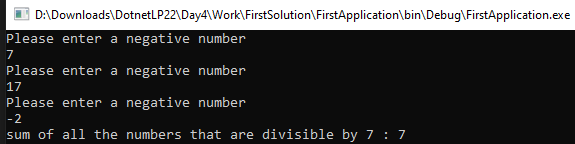
if (list[j] % 7 == 0)

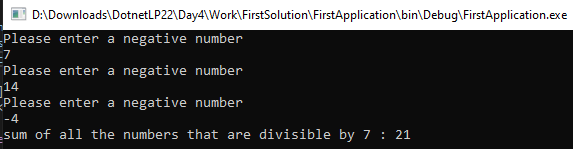
sum += list[j];

}

Console.WriteLine("sum of all the numbers that are divisible by 7 : "+sum);

}





9) Take a 4 digit number from user and find the sum of all the digits

example - 1234 result should be 10

public void SumOf4Digit()

{

int number, sum = 0;

string strNumber = "";

Console.WriteLine("Please enter a 4 digit number");

number = Convert.ToInt32(Console.ReadLine());

strNumber = number.ToString();

for (int i = 0; i < 4; i++)

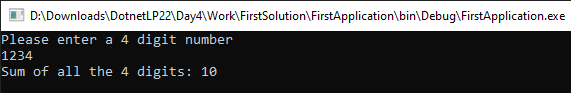
{

sum += Int32.Parse(strNumber[i].ToString());

}

Console.WriteLine("Sum of all the 4 digits: " + sum);

}



10) Take a 4 digit number from user and find if it is a palindrome or not

example - 1234 result should be Not a plaindrome

example - 1221 result should be Plaindrome

public void CheckPalindrome()

{

int number;

Console.WriteLine("Please enter a 4 digit number");

number = Convert.ToInt32(Console.ReadLine());

if (number < 0)

Console.WriteLine("Not Palindrome");

List<int> list = new List<int>();

while (number > 0)

{

list.Add(number % 10);

number = number / 10;

}

int ansLresult = 0, ansRresult = 0;

for (int i = 0; i < 2; i++)

{

//front

ansLresult = 10 \* ansLresult + list[i];

//Console.WriteLine("L:" + ansLresult);

}

for (int ii = list.Count - 1; ii >= 2; ii--)

{

//back / reversed

ansRresult = 10 \* ansRresult + list[ii];

//Console.WriteLine("R:" + ansRresult);

}

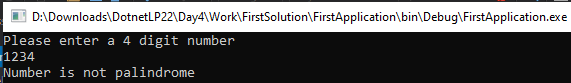
if(ansLresult == ansRresult)

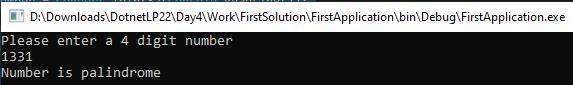
Console.WriteLine("Number is palindrome");

else

Console.WriteLine("Number is not palindrome");

}





11) https://leetcode.com/problems/powx-n/

public void GetPowerOfAnswer()

{

float x, n, total=1;

Console.WriteLine("For Pow(x, n), Please enter the x number");

x = float.Parse(Console.ReadLine());

Console.WriteLine("For Pow(x, n), Please enter the n number");

n = float.Parse(Console.ReadLine());

for (float i = 0; i < Math.Abs(n); i++)

{

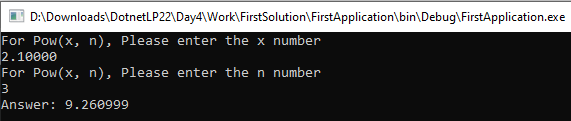
total \*= x;

}

if (n < 0)

total = 1 / total;

Console.WriteLine("Answer: " + total);

}

12) <https://leetcode.com/problems/happy-number/>

public void IsHappy()

{

int number;

string strNumber = "";

Boolean toRepeat = true;

Console.WriteLine("Please enter a number");

number = Convert.ToInt32(Console.ReadLine());

if (number<10)

{

Console.WriteLine("Number is not happy");

}

strNumber = number.ToString();

int noToSplitInto, sum=0;

noToSplitInto = strNumber.Length;

while (toRepeat)

{

for (int ii = 0; ii < noToSplitInto; ii++)

{

//Console.WriteLine(int.Parse(strNumber[ii].ToString()));

sum += int.Parse(strNumber[ii].ToString()) \* int.Parse(strNumber[ii].ToString());

//Console.WriteLine("each sum " + sum);

}

//last step

if (sum == 1)

{

Console.WriteLine("Number is happy");

toRepeat = false;

}

//convert sum into to string and break again

strNumber = sum.ToString();

noToSplitInto = strNumber.Length;

sum = 0;

}

}

