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 qn1()

{

int num;

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

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

Console.WriteLine("Result: ");

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

{

Console.WriteLine(i);

}

}

Text

Description automatically generated

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

public void qn2()

{

int num;

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

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

Console.WriteLine("Result: ");

if (num % 2 == 0)

{

Console.WriteLine(num + "is an Even number ");

}

else

Console.WriteLine(num + "is an Odd number");

}

Graphical user interface, text, application

Description automatically generated

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

public void qn3()

{

int num1, num2;

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

num1 = Convert.ToInt32(Console.ReadLine());

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

num2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Result: ");

if (num1 > num2)

{

Console.WriteLine(num1 + " is greater than " + num2);

}

else if (num1 < num2)

{

Console.WriteLine(num2 + " is greater than " + num1);

}

else

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

}

Graphical user interface, text, application

Description automatically generated

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

public void qn4()

{

int num1, num2, num3, greatest;

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

num1 = Convert.ToInt32(Console.ReadLine());

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

num2 = Convert.ToInt32(Console.ReadLine());

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

num3 = Convert.ToInt32(Console.ReadLine());

greatest = num1;

if (greatest < num2 && num2 > num3)

{

greatest = num2;

}

else if (greatest < num3 && num3 > num2)

greatest = num3;

Console.WriteLine("Result: ");

Console.WriteLine("The greatest number is " + greatest);

}

Text

Description automatically generated

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

public void qn5()

{

int num1; int num2;

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

num1 = Convert.ToInt32(Console.ReadLine());

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

num2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Result: ");

for (int i = num1 + 1; i < num2; i++)

{

Console.WriteLine(i);

}

}

Text

Description automatically generated

6) Find if a given number is prime

public void qn6()

{

int num1;

bool isPrime = true;

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

num1 = Convert.ToInt32(Console.ReadLine());

if (num1 <= 1)

{

isPrime = false;

}

for (int i = 2; i < num1; i++)

{

if (num1 % i == 0)

{

isPrime = false;

break;

}

}

Console.WriteLine("Result: ");

if (isPrime)

Console.WriteLine(num1 + " is a prime number");

else

Console.WriteLine(num1 + " is not a prime number");

}

Graphical user interface

Description automatically generated with medium confidence

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

public void qn7()

{

int num1, num2, count = 0;

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

num1 = Convert.ToInt32(Console.ReadLine());

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

num2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Result: ");

Console.WriteLine("Prime numbers between " + num1 + " and " + num2 + " are : ");

for (int i = num1; i < num2; i++)

{

count = 0;

if (i > 1)

{

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

{

if (i % j == 0)

{

count = 1;

break;

}

}

if (count == 0)

{

Console.WriteLine(i);

}

}

}

}

Text

Description automatically generated

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

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

example - 1234 result should be 10

public void qn9()

{

int n, sum = 0, m;

Console.WriteLine("Please Enter a number with 4 digits: ");

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

while (n > 0)

{

m = n % 10;

sum = sum + m;

n = n / 10;

}

Console.WriteLine("Result: ");

Console.WriteLine("Total of all digits are :" + sum);

}

Text

Description automatically generated

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 qn10()

{

int n, r, sum = 0, temp;

Console.WriteLine("Please enter a number with 4 digits: ");

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

temp = n;

while (n > 0)

{

r = n % 10;

sum = (sum \* 10) + r;

n = n / 10;

}

Console.WriteLine("Result: ");

if (temp == sum)

Console.WriteLine( temp + " is Palindrome.");

else

Console.WriteLine(temp + " is not Palindrome");

}

Graphical user interface, text

Description automatically generated

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

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

using System;

{

public class Program

{

public bool IsHappy(double n) {

if(n == 1)

return true;

else

return false;

}

public static void Main()

{

Console.WriteLine("Input : n = ");

var number1 = Console.ReadLine();

double m = Double.Parse(number1);

Program ma = new Program();

if(number1.Length == 1)

{

ma.IsHappy(m);

}

else

{

do

{

m = 0.00;

for (int i = 0; i < number1.Length; i++)

{

double[] numArray = new double[number1.Length];

numArray[i] = Double.Parse(number1[i].ToString());

m += Math.Pow(numArray[i], 2);

}

number1 = m.ToString();

} while (number1.Length != 1);

m = Double.Parse(number1);

ma.IsHappy(m);

}

}

}

1 Lim Wei Sin  Accessibility https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/#accessibility

2 Johanna Then Lasmauli  Sitompul Methods in (C#) https://docs.microsoft.com/en-us/dotnet/csharp/methods

3 Chang Li Pin Struct Instances vs. Class Instances https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/objects#struct-instances-vs-class-instances

4 Phoebe Toh Yong Yee Constructors https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/constructors

5 ESTER DING HOON SHING Partial Classes https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/partial-classes-and-methods#partial-classes

6 Ooi Shu Jing  Views in SQL https://docs.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver15

7 Chee Keh Keng General Structure of a C# Program https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/program-structure/

8 Abdul Halim bin Abdul Rahman The C# type system https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/types/

9 Mohammmad Afiq Bin Mohammad Amin Identifier names https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/identifier-names