**Question 1**

public void ZeroToNum()

{

Console.Write("Enter a number: ");

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

int i = 0;

while ( i < Math.Abs(num)+1 )

{

if ( num < 0 ) Console.WriteLine(-i);

else Console.WriteLine(i);

i++;

}

}

Text

Description automatically generated

**Question 2**

public void OddEven()

{

Console.Write("Enter a number: ");

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

int remainder = num % 2;

if ( remainder == 0 )

Console.WriteLine("The number " + num + " is even");

else

Console.WriteLine("The number " + num + " is odd");

}

**Graphical user interface, text

Description automatically generated**

**Question 3**

public void FindGreatest()

{

Console.Write("Enter first number: ");

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

Console.Write("Enter second number: ");

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

if (num1 > num2)

Console.WriteLine("The number " + num1 + " is the greatest");

else

Console.WriteLine("The number " + num2 + " is the greatest");

}

**Text

Description automatically generated**

**Question 4**

public void FindGreatest()

{

Console.Write("Enter first number: ");

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

Console.Write("Enter second number: ");

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

Console.Write("Enter third number: ");

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

if (num1 > num2 && num1 > num3)

Console.WriteLine("The number " + num1 + " is the greatest");

else if (num2 > num1 && num2 > num3)

Console.WriteLine("The number " + num2 + " is the greatest");

else

Console.WriteLine("The number " + num3 + " is the greatest");

}

**Graphical user interface, text

Description automatically generated**

**Question 5**

public void NumbersInBetween()

{

Console.Write("Enter first number: ");

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

Console.Write("Enter second number: ");

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

int min = num1;

int max = num2;

if (num1 > num2)

{

min = num2;

max = num1;

}

for ( int i = 0; i < max-min-1; i++)

{

Console.WriteLine(min+i+1);

}

}

**Text

Description automatically generated**

**Question 6**

static bool IsPrime(int number)

{

if (number <= 1) return false;

if (number == 2) return true;

if (number % 2 == 0) return false;

var boundary = (int)Math.Floor(Math.Sqrt(number));

for (int i = 3; i <= boundary; i += 2)

if (number % i == 0)

return false;

return true;

}

public void IsPrime()

{

Console.Write("Enter a number: ");

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

bool prime = IsPrime(num);

if ( prime == true )

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

else

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

}

**Graphical user interface, text

Description automatically generated**

**Question 7**

public void PrimeNumbersInBetween()

{

Console.Write("Enter first number: ");

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

Console.Write("Enter second number: ");

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

int min = num1;

int max = num2;

if (num1 > num2)

{

min = num2;

max = num1;

}

for (int i = 0; i < max - min - 1; i++)

{

int num = min + i + 1;

if ( IsPrime(num) == true )

Console.WriteLine(num);

}

}

**Text

Description automatically generated**

**Question 8**

public void DivBy7()

{

Console.Write("Enter a number: ");

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

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

int sum = 0;

while (input >= 0)

{

Console.Write("Enter another number: ");

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

numlist.Add(input);

}

foreach (int n in numlist)

{

if (n%7 == 0)

{

sum = sum + n;

}

}

Console.WriteLine("The sum of all the numbers entered that is divisible by 7 is " + sum);

}

**Text

Description automatically generated**

**Question 9**

public void SumOf4()

{

Console.Write("Enter a 4-digit number: ");

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

int sum = 0;

int num = input;

while (num != 0)

{

int remainder = num % 10;

sum = sum + remainder;

num = num / 10;

}

Console.WriteLine("The sum of digits in " + input + " is " + sum);

}

**A picture containing text, black, orange, dark

Description automatically generated**

**Question 10**

public void IsPalindrome()

{

Console.Write("Enter a 4-digit number: ");

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

bool ispal = false;

if (input / 1000 == input % 10)

if ((input / 100) % 10 == (input / 10) % 10)

ispal = true;

if (ispal == true)

Console.WriteLine("The number " + input + " is a palindrome");

else

Console.WriteLine("The number " + input + " is not a palindrome");

}

**Text

Description automatically generated**

**Question 11**

public double myPow(double x, int n)

{

double y = x;

for (int i=0; i<n-1; i++)

{

y = y \* x;

}

return y;

}

**Question 12**

static int SumOfSquaredDigits(int n)

{

int sum = 0;

int num = n;

while (num != 0)

{

int remainder = num % 10;

sum = sum + ( remainder \* remainder );

num = num / 10;

}

return sum;

}

public bool isHappy(int n)

{

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

numlist.Add(n);

int num = n;

while(true)

{

int newnum = SumOfSquaredDigits(num);

if (newnum == 1)

return true;

if (numlist.Contains(newnum) == true)

break;

num = newnum;

numlist.Add(num);

Console.WriteLine(num);

}

return false;

}