Assignment instructions

25 minutes to complete | 210 student solutions

Create a class called 'Number' with an instance field called 'value' that stores an integer value.

This class should contain the following methods to manipulate the number that is stored in the 'value' field.

```
class Number:
    - private int value //field
    - public void SetValue(int value)
    - public double GetValue()
    - public bool IsZero()
    - public bool IsPositive()
    - public bool IsNegative()
    - public bool IsOdd()
    - public bool IsEven()
    - public bool IsPrime()
    - public static int Power(int baseNumber, int exponent, bool recursive = false)
    - public int GetCountOfDigits()
    - public int GetSumOfDigits()
    - public int GetReverse()
    - public string ToWords()
    - public bool IsArmstrong()
    - public string GetFibonacci()
    - public bool isPalindrome()
```

The final output from Main method should be as follows:

```
    static void Main()
    {
        Number number = new Number();
        number.SetValue(371); //you can set any integer value
        System.Console.WriteLine("Value: " + number.GetValue()); //Output: 371
        System.Console.WriteLine("IsZero: " + number.IsZero()); //Output: False
        System.Console.WriteLine("IsPositive: " + number.IsPositive()); //Output: True
        System.Console.WriteLine("IsNegative: " + number.IsNegative());
```

```
        //Output: False
9           System.Console.WriteLine("IsOdd: " + number.IsOdd()); //Output:
        True
10          System.Console.WriteLine("IsEven: " + number.IsEven()); //Output:
        False
11          System.Console.WriteLine("IsPrime: " + number.IsPrime());
        //Output: False
12          System.Console.WriteLine("GetCountOfDigits: " +
        number.GetCountOfDigits()); //Output: 3
13          System.Console.WriteLine("GetSumOfDigits: " +
        number.GetSumOfDigits()); //Output: 11
14          System.Console.WriteLine("GetReverse: " + number.GetReverse());
        //Output: 173
15          System.Console.WriteLine("ToWords: " + number.ToWords());
        //Output: Three Seven One
16          System.Console.WriteLine("IsArmstrong: " + number.IsArmstrong());
        //Output: True
17          System.Console.WriteLine("GetFibonacci: " +
        number.GetFibonacci()); //Output: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,
        144, 233
18          System.Console.WriteLine("isPalindrome: " +
        number.isPalindrome()); //Output: False
19          System.Console.ReadKey();
20      }
```

Methods:

public void SetValue(int value)

`//Set method that receives a numeric value as argument and assigns the same into the 'value' field`

public double GetValue()

`//Get method that returns the numeric value of the current instance (object)`

public bool IsZero()

`//A method that returns a boolean value that indicates whether the current numeric value is equal to zero or not`

public bool IsPositive()

`//A method that returns a boolean value that indicates whether the current numeric value is a positive number or not`

public bool IsNegative()

```
//A method that returns a boolean value that indicates whether the current numeric value is a negative number or not
```

public bool IsOdd()

```
//A method that returns a boolean value that indicates whether the current numeric value is an odd number or not
```

public bool IsEven()

```
//A method that returns a boolean value that indicates whether the current numeric value is an even number or not
```

public bool IsPrime()

```
//A method that returns a boolean value that indicates whether the current numeric value is a prime number or not. If the number is not divisible by any other number (except '1' and same number itself) with remainder zero (0), it is a prime number. Eg: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97 etc., are the prime numbers. Because, for example, the number 7 is divisible by 1 and 7 only - but not by other numbers between, such as 2, 3, 4, 5, 6.
```

public static int Power(int baseNumber, int exponent, bool recursive = false)

```
1  //A mthod that returns power value of given baseNumber. "baseNumber power exponent". Eg: baseNumber is "2". exponent is "3". So the method should return the value of "2 power 3" (2 * 2 * 2) = 8.
2  //If the "recursive = true", this method should find the power value using 'recursion' technique; otherwise, using normal 'for' loop.
```

public int GetCountOfDigits()

```
//A method that returns the count of digits in a number. Eg: If the number is "98765", it has "5" digits. So the method returns '5'.
```

public int GetSumOfDigits()

```
//A method that returns sum of all digits in a number. Eg: If the
```

`number is "1234", it should calculate "1+2+3+4". So it's "10".`

public int GetReverse()

`//A method that returns reverse of the number. Eg: If the number is "1234", its reverse is "4321".`

public string ToWords()

`//A method that returns all digits of the number, as words. Eg: If the number is "9840", then the same number in words is "Nine Eight Four Zero".`

public static string GetWord(int digit)

`//A method that returns the given number in words. Eg: if the argument value is "1", it returns "One"`

public bool IsArmstrong()

```
1  /*A method that returns a boolean value that indicates whether the
   current numeric value is an Armstrong number or not.
2      Armstrong number is a number where the sum of its digits raised to
   the 'n' power is equal to the number itself, where the 'n' is the 'number
   of digits of the number'.
3      Eg: number is '371'. number of digits is "3" because the number has 3
   digits.
4      371 == (3*3*3) + (7*7*7) + (1*1*1)
5      371 == (27) + (343) + (1)
6      371 == 371
7  */
```

public string GetFibonacci()

```
1  /*A method that prints Fibonacci series from "0" to "current numeric
   value" i.e "value" field.
2  Eg:  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987,
   1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418,
   317811, ...
3
4  By default, we will consider minimum of two numbers i.e "0" and "1" in
   the series.
5  The next number is found by adding up the two numbers before it:
6  - the 1 is found by adding the two numbers before it (0+1),
7  - the 2 is found by adding the two numbers before it (1+1),
8  - the 3 is found by adding the two numbers before it (1+2),
```

```
- the 5 is (2+3),
- the 8 is (3+5),
and so on!
*/
```

public bool isPalindrome()

`//A method that returns true, if the number ('value' field) is a palindrome number`

Questions for this assignment

1. Check your source code with instructor's solution.