For tests 1 – 5, follow the instructions provided in the Submission.cs file.

## Test 1 – Write an overloaded constructor

Write an overloaded constructor that accepts a string (only). Use the string to initialize the member field, `greeting`. You do not need to update the value of any other member field(s).

## Test 2 – Write a 'Getter' method

Write a getter for the member field, `greeting`. The getter must be named `GetGreeting`. The getter will return the data type that matches the member field being returned and accept no parameters.

## Test 3 – Write a 'Setter' method

Write a setter for the member field, `greeting`. The setter must be named `SetGreeting`. The setter will not return anything and will accept a single parameter that matches the member field being updated.

## Test 4 – Write a C# property

Write a property, `Rendezvous`. The property will be a `public int` and will access/update the member field `rendezvous`

## Test 5 – Write a 'regular' method

Write a method named `GreetingMe` that accepts no parameters and returns the value of greeting (a string) as all capital letters - Use the `ToUpper` method

## Test 6 – Call a static method and cast the result

```
public static int Test6(int input)
```

Given an int, `input`, raise the input value to the 3rd power using the `Pow` method found in the Math class. Return the result as an int.

Example input

```
8
```

Example output

```
512
```

## Test 7 – Call a static method

```
public static double Test7(float input)
```

Given a float, `input`, find the square root of the input using the `Sqrt` method found in the Math class. Return the result.

Example input

```
98.81
```

Example output

```
9.9
```

## Test 8 – Use integer division to find a quotient

```
public static int Test8(int number1, int number2)
```

Given two int values, `number1` and `number2`, find the integer quotient when dividing number1 by number2.

Example input

```
19, 3
```

Example output

```
6
```

## Test 9 – Use integer division to find a remainder

```
public static int Test9(int number1, int number2)
```

Given two int values, `number1` and `number2`, find the integer remainder when dividing number1 by number2.

Example input

```
47, 13
```

Example output

```
8
```

## Test 10 – Create a Random object using a seed, generate a ranged random number

```
public static int Test10(int min, int max, int seed)
```

Given three int values, `min`, `max` and `seed`, create a Random object passing the `seed` provided to the constructor for Random. Using the appropriate Next method, generate a random number between `min` (inclusive) and `max` (exclusive). Return the generated random number.

Example input

```
3, 19, 66
```

Example output

```
6
```