Convert int values to roman numerals looks like task that was solved many times.

So the best way to do it is to go straightforward to google:

c# number to roman numeral

https://stackoverflow.com/questions/7040289/converting-integers-to-roman-numerals

```
public static string ToRoman(int number)
{
   if ((number < 0) || (number > 3999)) throw new ArgumentOutOfRangeException("insert value betwheen 1 and 3999");
   if (number < 1) return string.Empty;
   if (number >= 1000) return "M" + ToRoman(number - 1000);
   if (number >= 900) return "CM" + ToRoman(number - 900);
   if (number >= 500) return "D" + ToRoman(number - 500);
   if (number >= 400) return "CD" + ToRoman(number - 400);
   if (number >= 100) return "C" + ToRoman(number - 100);
   if (number >= 90) return "XC" + ToRoman(number - 90);
   if (number >= 50) return "L" + ToRoman(number - 50);
   if (number >= 40) return "XL" + ToRoman(number - 40);
   if (number >= 10) return "X" + ToRoman(number - 10);
   if (number >= 9) return "IX" + ToRoman(number - 9);
   if (number >= 5) return "V" + ToRoman(number - 5);
   if (number >= 4) return "IV" + ToRoman(number - 4);
   if (number >= 1) return "I" + ToRoman(number - 1);
   throw new ArgumentOutOfRangeException("something bad happened");
}
```

So, the first approach is very straightforward and easy to read, but can find something with better code quality.

Went through a lot of different approaches and found my personal favorite –

```
public static List<string> romanNumerals = new List<string>() { "M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I" };
   public static List<int> numerals = new List<int>() { 1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1 };

   public static string ToRomanNumeral(int number)
   {
      var romanNumeral = string.Empty;
      while (number > 0)
      {
         // find biggest numeral that is less than equal to number
         var index = numerals.FindIndex(x => x <= number);
         // subtract it's value from your number
         number -= numerals[index];
         // tack it onto the end of your roman numeral
         romanNumeral += romanNumerals[index];
      }
      return romanNumeral;
   }
```

This is also straightforward, easy to read and code quality looks good. I think this approach allows to convert not only to roman numerals, but also to other specific numerals with the same converting logic. So, instead of declaring two lists with roman numerals and int numerals in the same class with logic that generates roman string, I created separate interface `INumeralMapper` where are declared two lists: <string> and <int> with public get. Then, from this interface I created class `RomanNumeralMapper` where I set up values for public get for specific (roman) numerals and list with int values for mapping.

From given in task interface I created `RomanNumeralGenerator` where method Generate(int number) is implemented. One of this class fields is INumerallMapper Mapper, which is some kind of unification. In that case it would be logic also to rename interface given in task from `IRomanNumeralGenerator` to `ISpecificNumeralGenerator` but since it is given as a task, will leave it as it is. Result:

```
static void Main(string[] args)
{
    RomanNumeralGenerator generator = new RomanNumeralGenerator();
    generator.Generate(21);
    Console.WriteLine(generator.Generate(21));
}
```

```
XXI
```

Then, I added field MaxNumber and used it in validating number, and constructor for RomanNumeralGenerator where MaxNumber can be set (if passed without MaxNumber then default value is 3999).

Now need to create unit test for it and find some int-to-roman data. Go to google:

number roman numerals list to 1000

Found a good resource that seem to the one that can be trusted:

http://romannumerals.babuo.com/roman-numerals-1-1000

Changed 1000 in URL to 4000 and get the result:

http://romannumerals.babuo.com/roman-numerals-1-4000

Copy data from here to text file:

1: I

2: II

3: III

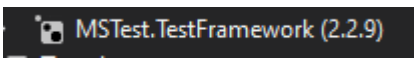
…

3999: MMMCMXCIX

Now to unit test:

Found good example in Microsoft docs:

https://docs.microsoft.com/en-us/visualstudio/test/walkthrough-creating-and-running-unit-tests-for-managed-code?view=vs-2022

Need to use Test framework for this: 

Created new MSTest project, added reference to RomanNumerals project, now I can use namespaces from RomanNumerals project.

Now, to the test itself. Need to read text file as list, line by line. Google:

c# read file line by line

https://www.techiedelight.com/read-file-line-by-line-csharp/

Just copy two lines of code from there in created `TestCreateRomanNumeral`:

```
string fileName = @"C:\some\path\file.txt";

    IEnumerable<string> lines = File.ReadLines(fileName);
```

Create again two lists, store spliited lines there (splitted number in a number list and romanNumerals in roman list). Then, generate romanNumerals using my own classes by passing list with numbers there and compare my generated roman numerals with romanNumerals from http://romannumerals.babuo.com/roman-numerals-1-4000

(Assert if they are not equal)

Test is passing