Description:

String Calculator Kata (via Roy Osherove)

- Create a simple String calculator with a method int Add(string numbers). The method can take 0, 1 or 2 numbers, and will return their sum (for an empty string it will return 0). For example "" or "1" or "1,2"
  - Start with the simplest test case of an empty string and move to 1 and two numbers
  - Remember to solve things as simply as possible so that you force yourself to write tests you did not think about
  - Remember to refactor after each passing test
- Allow the Add method to handle an unknown amount of numbers
- Allow the Add method to handle new lines between numbers (instead of commas).
  - the following input is ok: "1\n2,3" (will equal 6)
  - the following input is NOT ok: "1,\n" (not need to prove it - just clarifying)
- Support different delimiters. To change a delimiter, the beginning of the string will contain a separate line that looks like this: `[delimiter]\n[numbers...]`, for example `;\n1;2` should return three where the default delimiter is `;` .
  - he first line is optional. all existing scenarios should still be supported
- Calling Add with a negative number will throw an exception "negatives not allowed" - and the negative that was passed.
  - if there are multiple negatives, show all of them in the exception message
- Numbers bigger than 1000 should be ignored, so adding 2 + 1001 = 2
- Delimiters can be of any length with the following format: “//[delimiter]\n” for example: “//[***]\n1***2***3” should return 6
- Allow multiple delimiters like this: “//[delim1][delim2]\n” for example “//[*][%]\n1*2%3” should return 6.
- make sure you can also handle multiple delimiters with length longer than one char

Tasks:

1. calculate empty string:

    a. input "", return 0
    b. input null, return 0
2. calculate 1 number string
    a. input "1", return 1;
    b. input "123", return 123;
3. calculate 2 numbers string
    a. input "1,2", return 3
    b. input "123,1", return 124
4. calculate more than 2 numbers string
    a. input "0,1,456", return 457
5. calculate when numbers can be split by , or newLines
    a. input "0\n89", return 89
    b. input "0\n89,1", return 89
6. calculate when user specify the delimiter
    a. input ";\n0;1;1", return 2
7. showErrorMessage When There Are One Negative Number
    a. input "-1,1", throw an exception "negatives not allowed - -1"
8. showErrorMessage when there are more than one negative number
    a. input "-1,-1", throw an exception "negatives not

allowed - -1,-1"

   b. input "-1,2,-1" throw an exception "negatives not allowed - -1,-1"

9. number higher than 1000 should be ignored
   a. input '1,1000', return 1000
   b. input '1,1001',return 1
10. sum when user specify delimiters with any length in the format [delimiter]
   a. input '[***]\n1***2***3', return 6
11. sum when user specify multi delimiters with one character in the formart of [delimiter][delimiter]
   a. input '[*][%]\n2*5%1', return 8
12. sum when user specify multi delimiters with any length
   a. input '[***][;][,]\n2;2***5,1 return 10