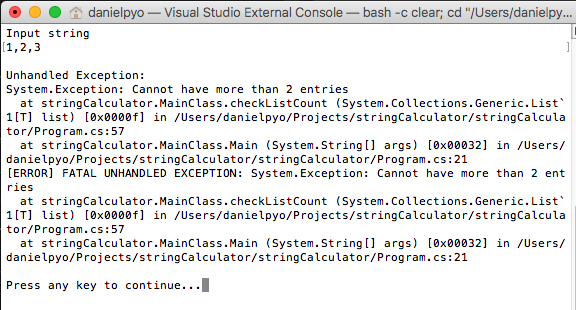
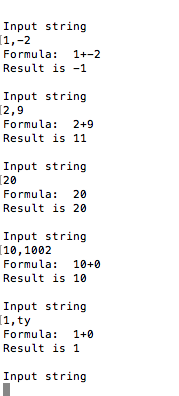
|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Example | Results |
| 1 | Support a maximum of 2 numbers using a comma delimiter. Throw an exception when more than 2 numbers are provided | 20 will return 20; 1,5000 will return 5001; 4,-3 will return 1 empty input or missing numbers should be converted to 0 invalid numbers should be converted to 0 e.g. 5,tytyt will return 5 | As expected |
| 2 | Remove the maximum constraint for numbers | 1,2,3,4,5,6,7,8,9,10,11,12 will return 78 | As expected |
| 3 | Support a newline character as an alternative delimiter | 1\n2,3 will return 6 | As expected |
| 4 | Deny negative numbers by throwing an exception that includes all of the negative numbers provided | -1,-4,ty\n4,1 | As expected |
| 5 | Make any value greater than 1000 an invalid number | 2,1001,6 will return 8 | As expected |
| 6 | Support 1 custom delimiter of a single character using the format: //{delimiter}\n{numbers} | //#\n2#5 will return 7; //,\n2,ff,100 will return 102 | As expected |
| 7 | Support 1 custom delimiter of any length using the format: //[{delimiter}]\n{numbers} | //[\*\*\*]\n11\*\*\*22\*\*\*33 will return 66 | As expected |
| 8 | Support multiple delimiters of any length using the format: //[{delimiter1}][{delimiter2}]...\n{numbers} | //[\*][!!][r9r]\n11r9r22\*hh\*33!!44 will return 110 | As expected |

**Test Case 1:**

Support a maximum of 2 numbers using a comma delimiter. Throw an exception when more than 2 numbers are provided



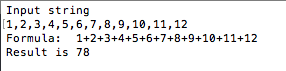
Additional Input:



Results are as expected

**Test Case 2:**

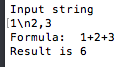
Remove the maximum constraint for numbers



Results are as expected

**Test Case 3:**

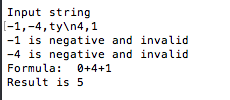
Support a newline character as an alternative delimiter



Results are as expected

**Test Case 4**

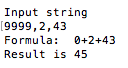
Deny negative numbers by throwing an exception that includes all of the negative numbers provided



Results are as expected

**Test Case 5:**

Make any value greater than 1000 an invalid number

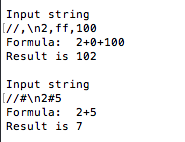


9999 is regarded as zero.

Results are as expected

**Test Case 6:**

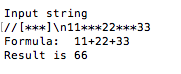
Support 1 custom delimiter of a single character using the format: //{delimiter}\n{numbers}



Results are as expected

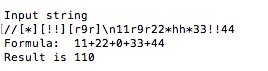
**Test Case 7:**

Support 1 custom delimiter of any length using the format: //[{delimiter}]\n{numbers}



Results are as expected

**Test Case 8:**

Support multiple delimiters of any length using the format: //[{delimiter1}][{delimiter2}]...\n{numbers}

Results are as expected