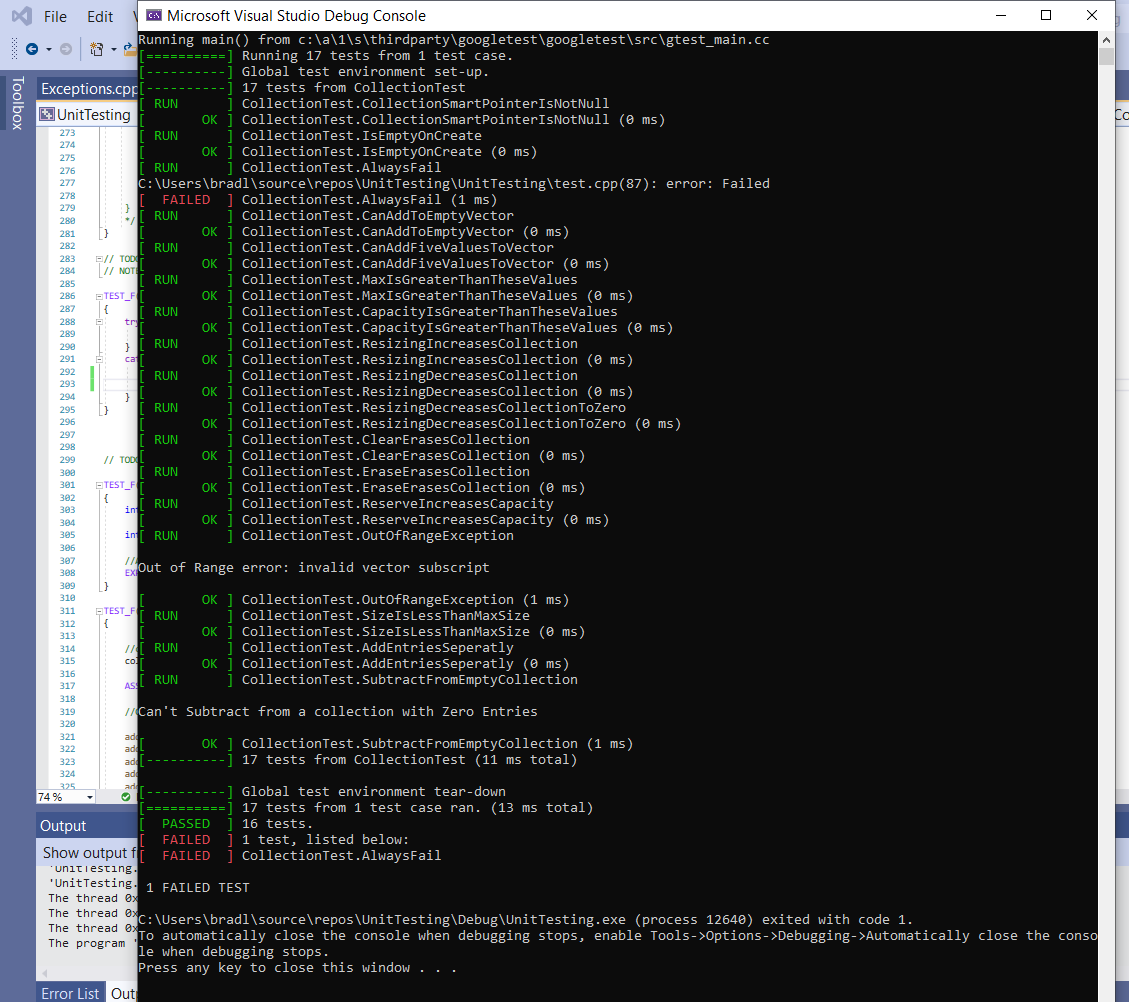
**Final Output**



All tests were passed except the “always fail” test.

I will explain more details below

**Testing breakdowns by Test name (in order of operation)**

*CanAddToEmptyVector*

* Empty collection, verify size is zero, add 1 entry, Assert that it’s false that collection is empty, which means there is something in collection (not zero), Then assert collection size is equal to 1

*CanAddFiveValuesToVector*

* Add 5 entries to collection, verify collection is not empty, then verify collection size is equal to 5.

*MaxIsGreaterThanTheseValues*

* Verify that it is true that max size is greater than the given values for size

*CapacityIsGreaterThanTheseValues*

* If the collection size is zero, verify capacity of the collection is greater than the size of the collection. Repeat for each size.
* An additional idea I had is commented in also, that I didn’t use.

*ResizingIncreasesCollection)*

* Struggled with this. I assumed I was suppose to check size before and after to verify resize increased collection size. I thought I could save collection size as a variable then compare it after I resized.
* I left some comments of things I was trying
* New Idea for code
  + Resized bigger and verify size increases
  + collection->resize(5);

ASSERT\_EQ(collection->size(),5);

Collection->resize(6);

ASSERT\_EQ(collection->size(),6);

*ResizingDecreasesCollection*

* similar to above methods, new code shows sized getting decreased
* New Idea for code
  + collection->resize(5);

ASSERT\_EQ(collection->size(),5);

Collection->resize(2);

ASSERT\_EQ(collection->size(),2);

*ResizingDecreasesCollectionToZero*

* re-size collection to a size of zero, then verify collection is empty, if empty it’s size is zero.

*ClearErasesCollection*

* use clear keyword, then verify collection size is zero

*EraseErasesCollection*

* I could not figure out how to incorporate erase method.

*ReserveIncreasesCapacity*

* Saved capacity and size into variables, before and after using reserve keyword, then compared them to show size didn’t change, but capacity did increase
* Didn’t know what reserve did so I looked it up and included comments of its code

*OutOfRangeException*

* Use try catch exception handling here. Used at to define a collection range that was too high, then catch it as an out of range error, and display that info to the screen.

*SizeIsLessThanMaxSize*

* My custom test 1
  + Save size and max size as variables.
  + We expect the max size to be higher than the size

*AddEntriesSeperatly*

* My custom test 2
  + Here I used the add entries method we used earlier, but added entries seperatly and not all at once, totaling 5 entries
  + Then I verified the size was 5

*SubtractFromEmptyCollection*

* My custom test 3 (negative)
  + Added one additional test
  + Cleared collection (size zero)
  + Tried to resize collection to a negative size
  + Used try catch to catch exception
  + Make sure size is zero
    - Size should stay zero which shows no operation was done on this collection
  + Tell user you can’t subtract from a collection of size zero.