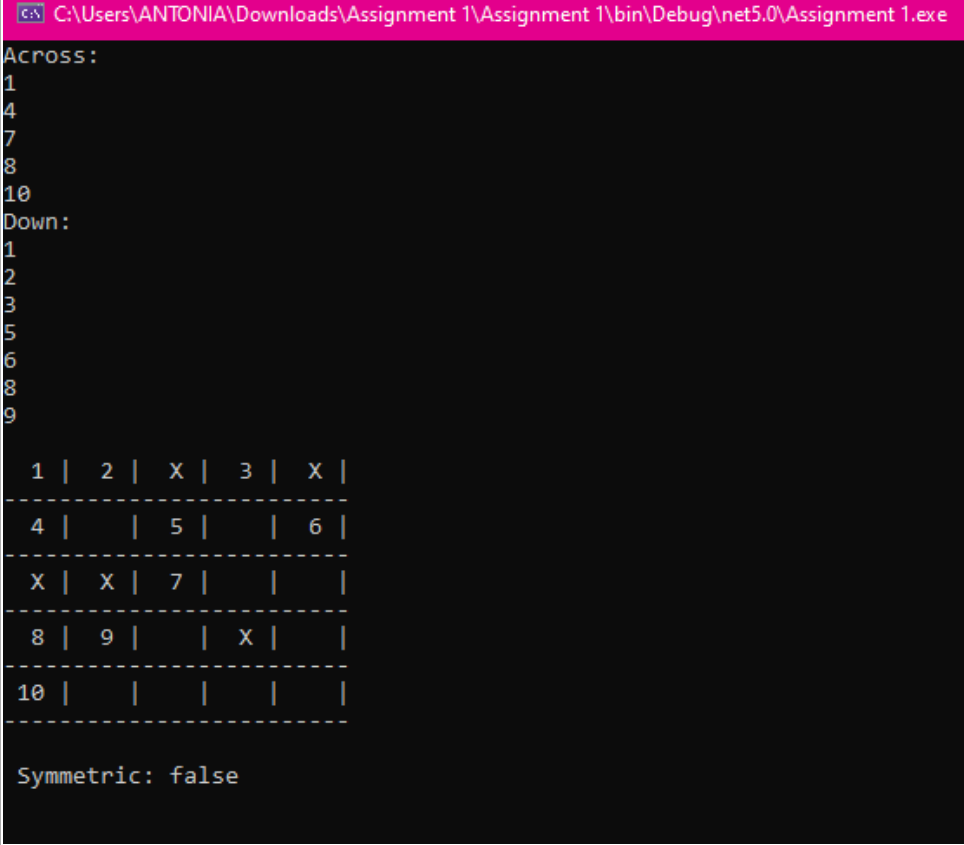


Assignment 1 Testing Documentation

Name: Dylan Brosseau (0724989)
Luka Piplica (0748533)
Opeyemi Okunmuyide (0736535)

Test 1	
Description	Using a 5 by 5 grid with 5 black boxes
Main()	<pre>Puzzle puz = new Puzzle(5); puz.Initialize(5); puz.Number(); puz.PrintClues(); puz.PrintGrid(); bool s = puz.Symmetric(); if (s == true) Console.WriteLine("\n Symmetric: true"); else Console.WriteLine("\n Symmetric: false");</pre>
Expected Output	A 5 by 5 grid with 5 randomly placed black boxes and correct lists of clues

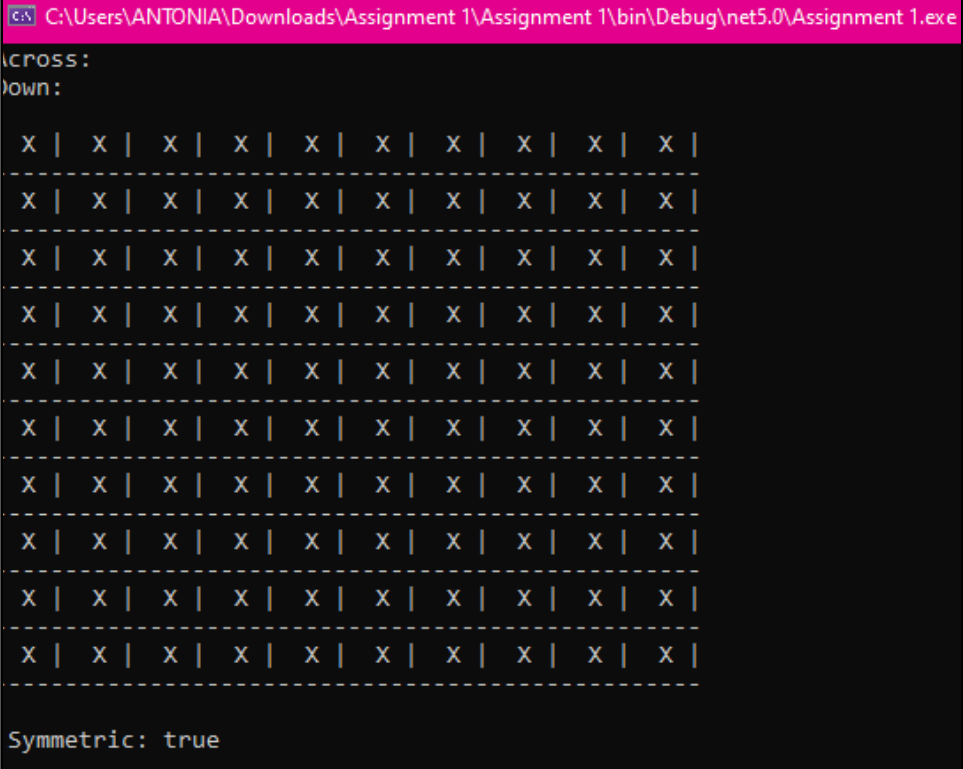
Actual Output	 <pre>C:\Users\ANTONIA\Downloads\Assignment 1\Assignment 1\bin\Debug\net5.0\Assignment 1.exe Across: 1 4 7 8 10 Down: 1 2 3 5 6 8 9 1 2 X 3 X ----- 4 5 6 ----- X X 7 ----- 8 9 X ----- 10 ----- Symmetric: false</pre>
---------------	---

Test 2	
Description	Using 10 by 10 grid and having a greater number of black boxes
Main()	<pre>Puzzle puz = new Puzzle(10); puz.Initialize(30); puz.Number(); puz.PrintClues(); puz.PrintGrid();</pre>

	<pre>bool s = puz.Symmetric(); if (s == true) Console.WriteLine("\n Symmetric: true"); else Console.WriteLine("\n Symmetric: false");</pre>
Expected Output	A 10 by 10 grid with 30 randomly placed boxes and correct clue lists
Actual Output	

	<pre> 1 2 3 X 4 X X X 5 6 ----- 7 8 9 10 ----- 11 X X 12 X X ----- X X 13 14 15 16 ----- 17 X X X 18 ----- X X X 19 X X X X ----- 20 21 22 23 X X 24 ----- 25 X X 26 ----- X 27 X X 28 X ----- 29 X X 30 ----- Symmetric: false </pre>
--	---

Test 3	
Description	Using a 10 by 10 grid filled with black boxes
Main()	Puzzle puz = new Puzzle(10); puz.Initialize(100); puz.Number(); puz.PrintClues(); puz.PrintGrid(); bool s = puz.Symmetric(); if (s == true) Console.WriteLine("\n Symmetric: true");

	<pre>else Console.WriteLine("\n Symmetric: false");</pre>
Expected Output	A 10 by 10 grid full of black boxes with no clues
Actual Output	 <pre>C:\Users\ANTONIA\Downloads\Assignment 1\Assignment 1\bin\Debug\net5.0\Assignment 1.exe ACROSS: Down: X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- X X X X X X X X X X ----- Symmetric: true</pre>

Test 4	
Description	Using 10 by 10 grid with no black boxes
Main()	<pre>Puzzle puz = new Puzzle(10); puz.Initialize(0); puz.Number();</pre>

	<pre>puz.PrintClues(); puz.PrintGrid(); bool s = puz.Symmetric(); if (s == true) Console.WriteLine("\n Symmetric: true"); else Console.WriteLine("\n Symmetric: false");</pre>
Expected Output	A 10 by 10 grid with no black boxes and 19 clues, with clue 1 going down and across

Actual Output

C:\Users\ANTONIA\Downloads\Assignment 1\Assignment 1\bin\Debug\net5.0\Assignment 1.exe

Across:

1

11

12

13

14

15

16

17

18

19

Down:

1

2

3

4

5

6

7

8

9

10

	1	2	3	4	5	6	7	8	9	10
--	---	---	---	---	---	---	---	---	---	----

11										
----	--	--	--	--	--	--	--	--	--	--

12										
----	--	--	--	--	--	--	--	--	--	--

13										
----	--	--	--	--	--	--	--	--	--	--

14										
----	--	--	--	--	--	--	--	--	--	--

15										
----	--	--	--	--	--	--	--	--	--	--

16										
----	--	--	--	--	--	--	--	--	--	--

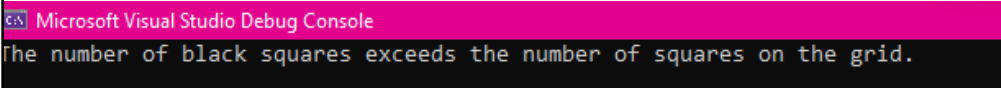
17										
----	--	--	--	--	--	--	--	--	--	--

18										
----	--	--	--	--	--	--	--	--	--	--

19										
----	--	--	--	--	--	--	--	--	--	--

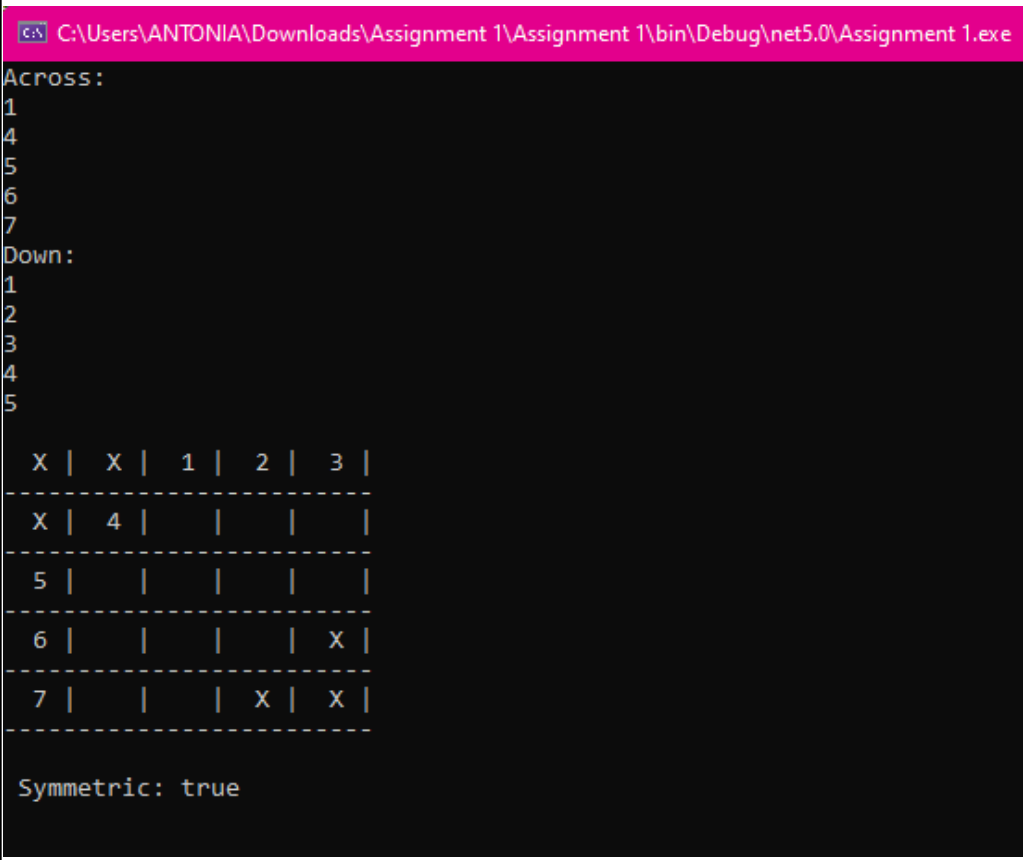
Symmetric: true

--	--

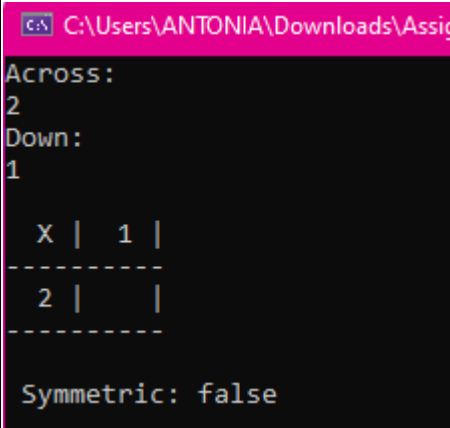
Test 5	
Description	Using a 10 by 10 grid with a number of black boxes that is greater than the grid size
Main()	<pre>Puzzle puz = new Puzzle(10); puz.Initialize(101); puz.Number(); puz.PrintClues(); puz.PrintGrid(); bool s = puz.Symmetric(); if (s == true) Console.WriteLine("\n Symmetric: true"); else Console.WriteLine("\n Symmetric: false");</pre>
Expected Output	An error message saying black boxes are more than grid size
Actual Output	

Test 6	
Description	Specifying grid boxes to get a symmetric grid in a 5 by 5 grid

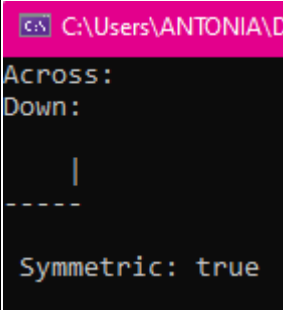
Main() / Initialize()	<pre>*This code is inside Initialize() and the code that randomly assigns black squares is commented out* grid[0, 0].Color = TColor.BLACK; grid[0, 1].Color = TColor.BLACK; grid[1, 0].Color = TColor.BLACK; grid[N - 1, N - 1].Colour = TColor.BLACK; grid[N - 1, N - 2].Color = TColor.BLACK; grid[N - 2, N - 1].Color = TColor.BLACK; * This code is inside Main() Puzzle puz = new Puzzle(5); puz.Initialize(0); puz.Number(); puz.PrintClues(); puz.PrintGrid(); bool s = puz.Symmetric(); if (s == true) Console.WriteLine("\n Symmetric: true"); else Console.WriteLine("\n Symmetric: false");</pre>
Expected Output	grid [0, 0], grid [0, 1], grid [1, 0], grid [4,4], grid [4, 3], grid [3, 4] are black squares and symmetric returns true.

Actual Output	 <pre>C:\Users\ANTONIA\Downloads\Assignment 1\Assignment 1\bin\Debug\net5.0\Assignment 1.exe Across: 1 4 5 6 7 Down: 1 2 3 4 5 X X 1 2 3 ----- X 4 ----- 5 ----- 6 X ----- 7 X X ----- Symmetric: true</pre>
---------------	---

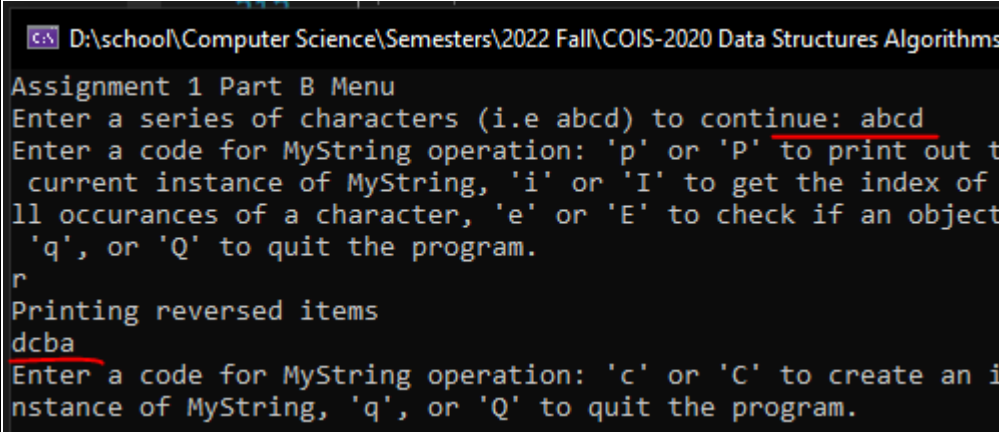
Test 7	
Description	Using a 2 by 2 grid
Main()	Puzzle puz = new Puzzle(2); puz.Initialize(1); puz.Number(); puz.PrintClues(); puz.PrintGrid();

	<pre> bool s = puz.Symmetric(); if (s == true) Console.WriteLine("\n Symmetric: true"); else Console.WriteLine("\n Symmetric: false"); </pre>
Expected Output	A 2 by 2 grid with one black square
Actual Output	 <pre> C:\Users\ANTONIA\Downloads\Assign... Across: 2 Down: 1 x 1 --- 2 --- Symmetric: false </pre>

Test 8	
Description	Using a 1 by 1 grid
Main()	<pre> Puzzle puz = new Puzzle(1); puz.Initialize(0); puz.Number(); puz.PrintClues(); puz.PrintGrid(); bool s = puz.Symmetric(); </pre>

	<pre>if (s == true) Console.WriteLine("\n Symmetric: true"); else Console.WriteLine("\n Symmetric: false");</pre>
Expected Output	A 1 by 1 grid with no black boxes
Actual Output	

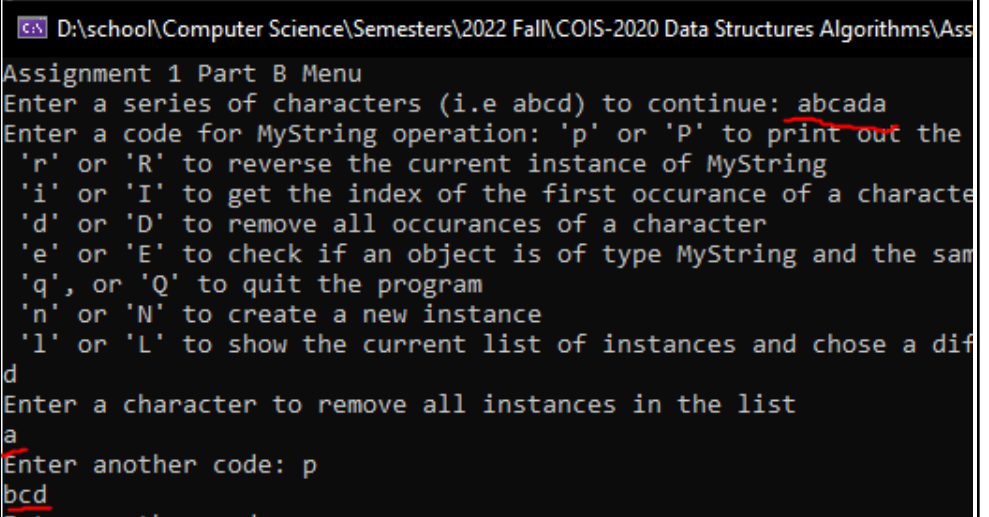
Part B - Tests

Test 1	
Description	Reverse the current instance of MyString
Input	abcd
Expected Output	dcba
Actual Output	 <pre> C:\D:\school\Computer Science\Semesters\2022 Fall\COIS-2020 Data Structures Algorithms Assignment 1 Part B Menu Enter a series of characters (i.e abcd) to continue: abcd Enter a code for MyString operation: 'p' or 'P' to print out the current instance of MyString, 'i' or 'I' to get the index of all occurrences of a character, 'e' or 'E' to check if an object exists, 'q', or 'Q' to quit the program. r Printing reversed items dcba Enter a code for MyString operation: 'c' or 'C' to create an i instance of MyString, 'q', or 'Q' to quit the program. </pre>

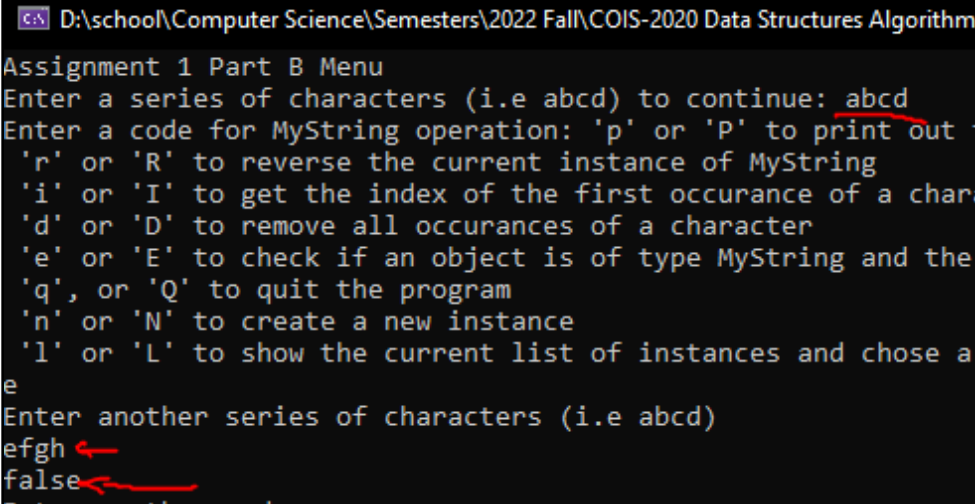
Test 2	
Description	Return the index of the first occurrence of c in this instance
Input	abcd - list input r - char c input
Expected Output	-1

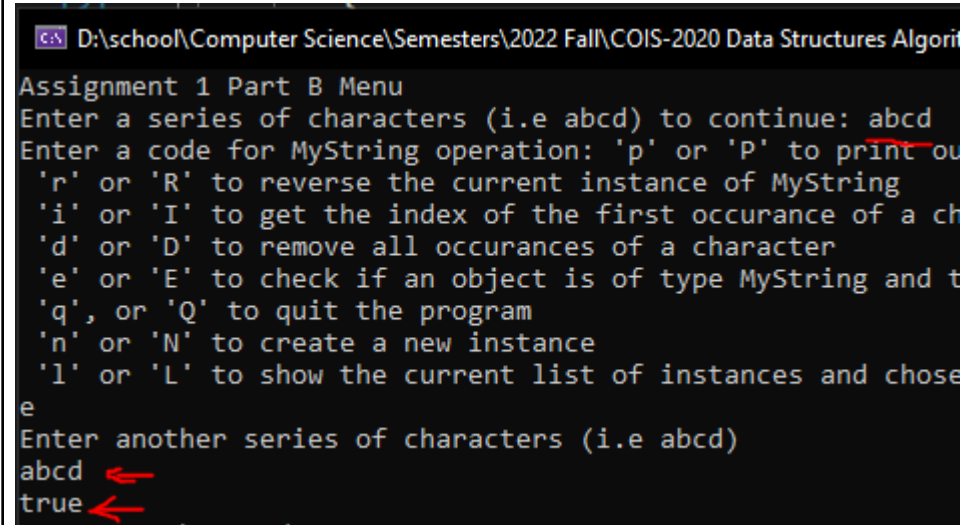
Actual Output	<pre> C:\D:\school\Computer Science\Semesters\2022 Fall\COIS-2020 Data Structures Algorithms\Assignments\Assignment 1\code\Assignment1 Assignment 1 Part B Menu Enter a series of characters (i.e abcd) to continue: abcd Enter a code for MyString operation: 'p' or 'P' to print out the current instance of MyString 'r' or 'R' to reverse the current instance of MyString 'i' or 'I' to get the index of the first occurrence of a character 'd' or 'D' to remove all occurrences of a character 'e' or 'E' to check if an object is of type MyString and the same as the current instance 'q', or 'Q' to quit the program 'n' or 'N' to create a new instance 'l' or 'L' to show the current list of instances and chose a different instance i Enter a char to find index in list c Result: -1 </pre>
---------------	--

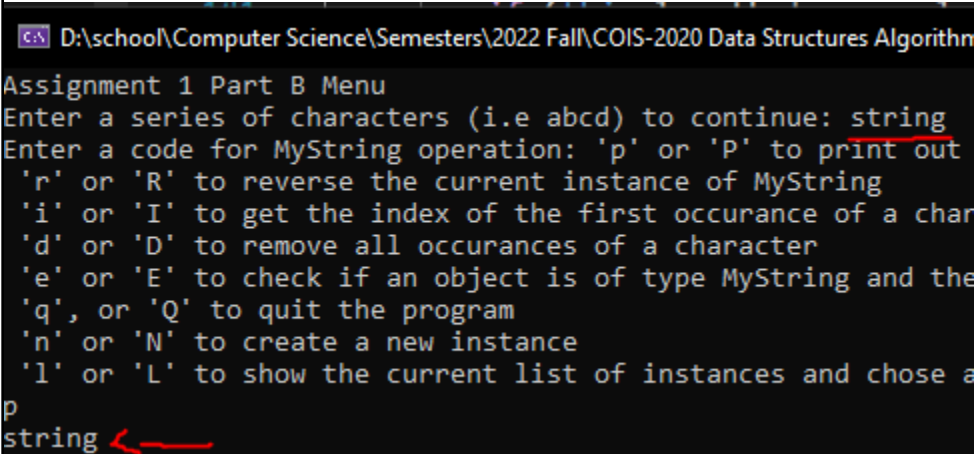
Test 3	
Description	Return the index of the first occurrence of c in this instance
Input	abcd - list c - char c input
Expected Output	2
Actual Output	<pre> C:\D:\school\Computer Science\Semesters\2022 Fall\COIS-2020 Data Structures Algorithms\Assign Assignment 1 Part B Menu Enter a series of characters (i.e abcd) to continue: abcd Enter a code for MyString operation: 'p' or 'P' to print out the c 'r' or 'R' to reverse the current instance of MyString 'i' or 'I' to get the index of the first occurrence of a character 'd' or 'D' to remove all occurrences of a character 'e' or 'E' to check if an object is of type MyString and the same 'q', or 'Q' to quit the program 'n' or 'N' to create a new instance 'l' or 'L' to show the current list of instances and chose a diff i Enter a char to find index in list c Result: 2 </pre>

Test 4	
Description	Remove all occurrences of c from this instance
Input	abcada - list input A - char c input
Expected Output	bcd
Actual Output	 <pre> C:\> D:\school\Computer Science\Semesters\2022 Fall\COIS-2020 Data Structures Algorithms\Ass Assignment 1 Part B Menu Enter a series of characters (i.e abcd) to continue: abcada Enter a code for MyString operation: 'p' or 'P' to print out the 'r' or 'R' to reverse the current instance of MyString 'i' or 'I' to get the index of the first occurrence of a character 'd' or 'D' to remove all occurrences of a character 'e' or 'E' to check if an object is of type MyString and the same 'q', or 'Q' to quit the program 'n' or 'N' to create a new instance 'l' or 'L' to show the current list of instances and chose a different d Enter a character to remove all instances in the list a Enter another code: p bcd </pre>

Test 5	
Description	Return true if obj is both of type MyString and the same as this instance
Input	abcd - first instance input efgh - second instance input
Expected Output	false

Actual Output	 <pre> D:\school\Computer Science\Semesters\2022 Fall\COIS-2020 Data Structures Algorithm Assignment 1 Part B Menu Enter a series of characters (i.e abcd) to continue: abcd Enter a code for MyString operation: 'p' or 'P' to print out 'r' or 'R' to reverse the current instance of MyString 'i' or 'I' to get the index of the first occurrence of a character 'd' or 'D' to remove all occurrences of a character 'e' or 'E' to check if an object is of type MyString and the 'q', or 'Q' to quit the program 'n' or 'N' to create a new instance 'l' or 'L' to show the current list of instances and chose a e Enter another series of characters (i.e abcd) efgh false </pre>
---------------	---

Test 6	
Description	Return true if obj is both of type MyString and the same as this instance
Input	abcd - first instance input abcd - second instance input
Expected Output	true
Actual Output	 <pre> D:\school\Computer Science\Semesters\2022 Fall\COIS-2020 Data Structures Algorithm Assignment 1 Part B Menu Enter a series of characters (i.e abcd) to continue: abcd Enter a code for MyString operation: 'p' or 'P' to print out 'r' or 'R' to reverse the current instance of MyString 'i' or 'I' to get the index of the first occurrence of a character 'd' or 'D' to remove all occurrences of a character 'e' or 'E' to check if an object is of type MyString and the 'q', or 'Q' to quit the program 'n' or 'N' to create a new instance 'l' or 'L' to show the current list of instances and chose e Enter another series of characters (i.e abcd) abcd true </pre>

Test 7	
Description	Print out this instance of MyString
Input	string - instance input
Expected Output	string
Actual Output	 <pre>C:\> D:\school\Computer Science\Semesters\2022 Fall\COIS-2020 Data Structures Algorithms Assignment 1 Part B Menu Enter a series of characters (i.e abcd) to continue: string Enter a code for MyString operation: 'p' or 'P' to print out 'r' or 'R' to reverse the current instance of MyString 'i' or 'I' to get the index of the first occurrence of a character 'd' or 'D' to remove all occurrences of a character 'e' or 'E' to check if an object is of type MyString and the 'q', or 'Q' to quit the program 'n' or 'N' to create a new instance 'l' or 'L' to show the current list of instances and chose a p string ←</pre>