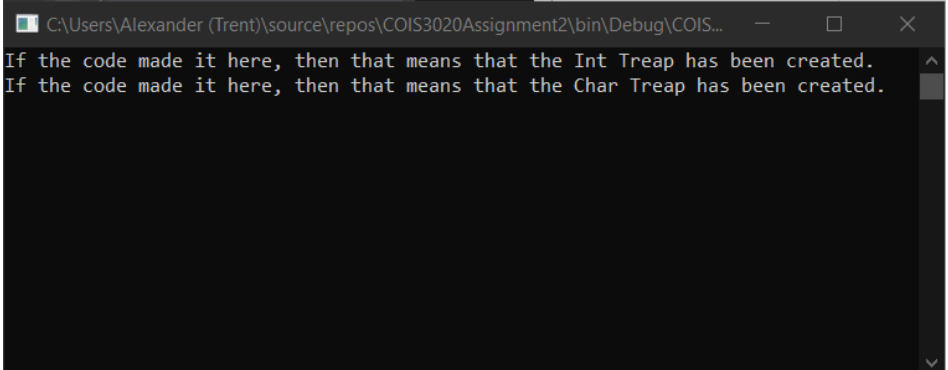
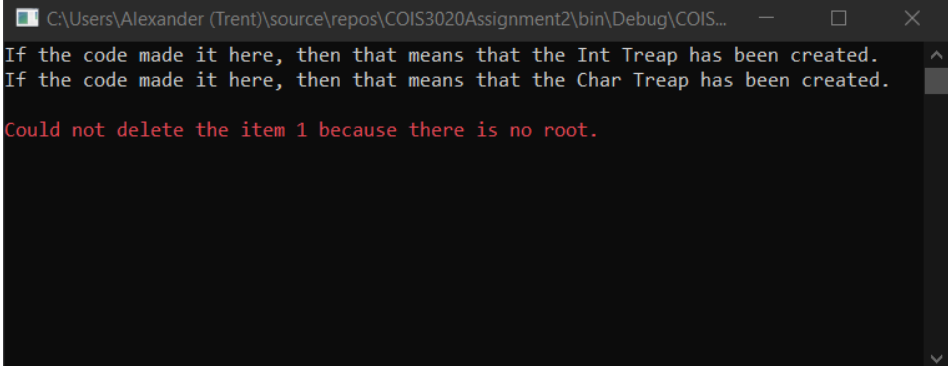


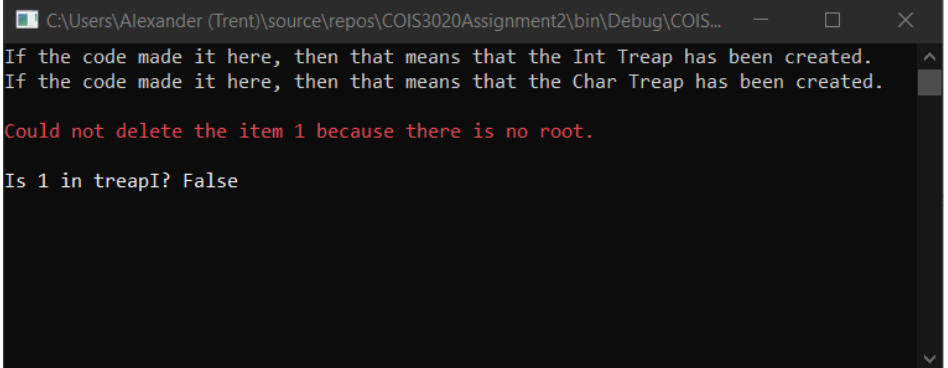
COIS 3020H: Data Structures and Algorithms II

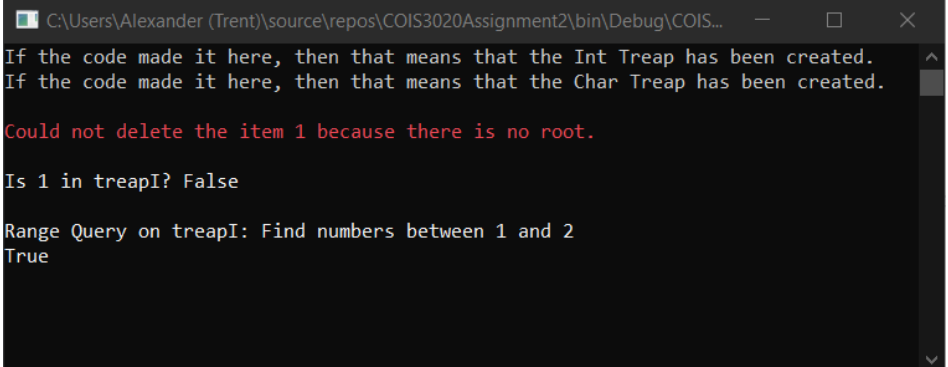
Assignment 2 Documentation and Analysis

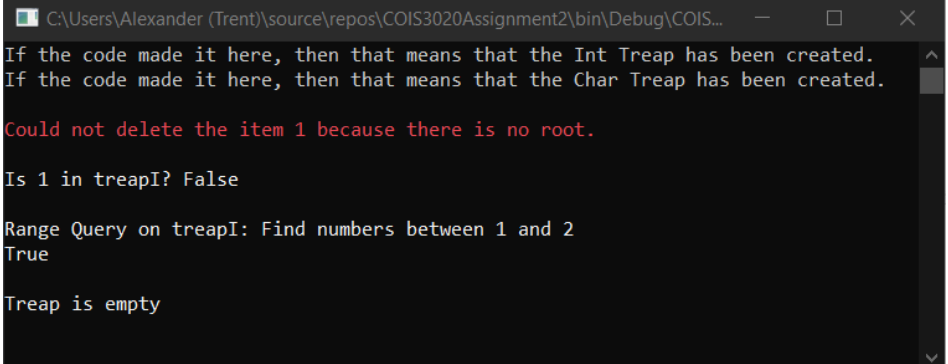
Due Date: November 20th, 2023

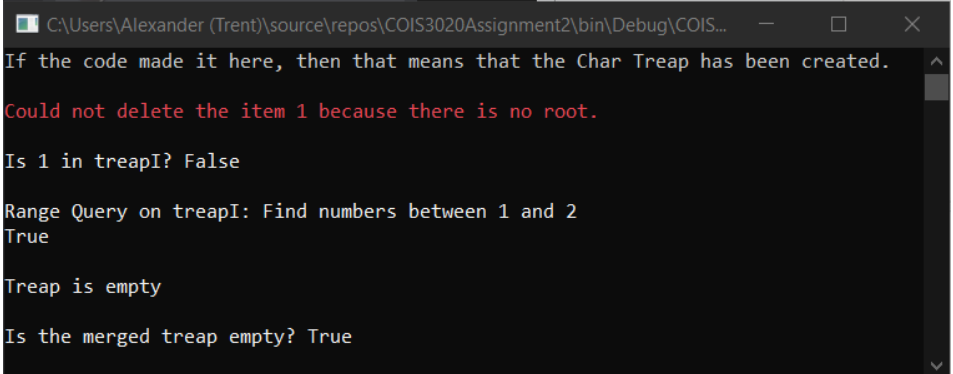
Case 1: Creating Treaps (Int and Char)	
Description	This case is to test if the Generic Treap Class can be created with multiple different data types without crashing.
Expected Output	If the code made it here, then that means that the Int Treap has been created. If the code made it here, then that means that the Char Treap has been created.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... If the code made it here, then that means that the Int Treap has been created. If the code made it here, then that means that the Char Treap has been created. </pre>

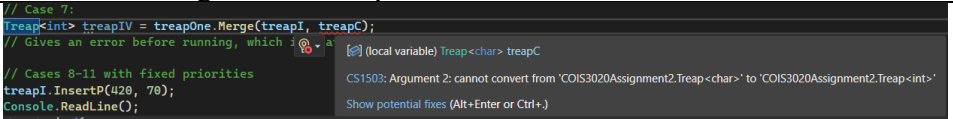
Case 2: Removing in an Empty Treap	
Description	This case is to test if removing in an Empty Treap fails immediately.
Expected Output	Could not delete the item 1 because there is no root.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... If the code made it here, then that means that the Int Treap has been created. If the code made it here, then that means that the Char Treap has been created. Could not delete the item 1 because there is no root. </pre>

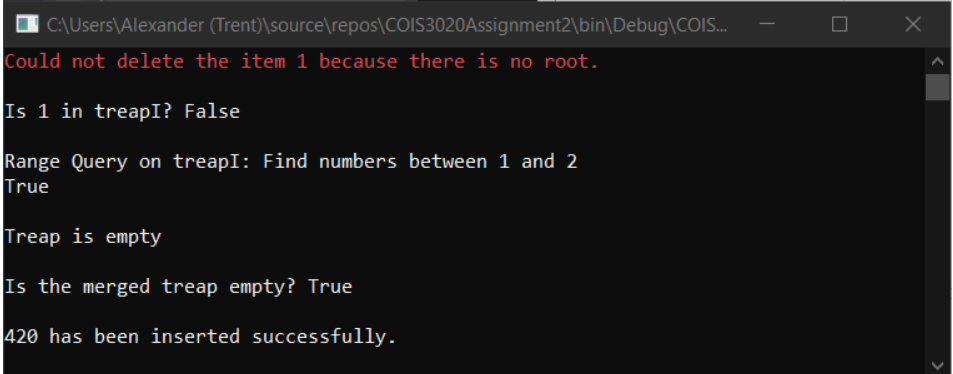
Case 3: Searching in an Empty Treap	
Description	This case is to test if searching in an Empty Treap returns false.
Expected Output	Is 1 in treapI? False
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... If the code made it here, then that means that the Int Treap has been created. If the code made it here, then that means that the Char Treap has been created. Could not delete the item 1 because there is no root. Is 1 in treapI? False </pre>

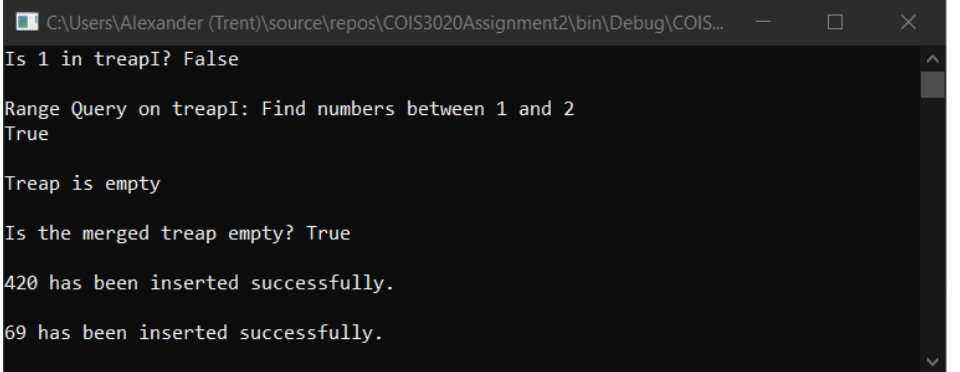
Case 4: Range Querying in an Empty Treap	
Description	This case is to test if range querying in an Empty Treap returns false.
Expected Output	Range Query on treapI: Find numbers between 1 and 2 True
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... If the code made it here, then that means that the Int Treap has been created. If the code made it here, then that means that the Char Treap has been created. Could not delete the item 1 because there is no root. Is 1 in treapI? False Range Query on treapI: Find numbers between 1 and 2 True </pre>

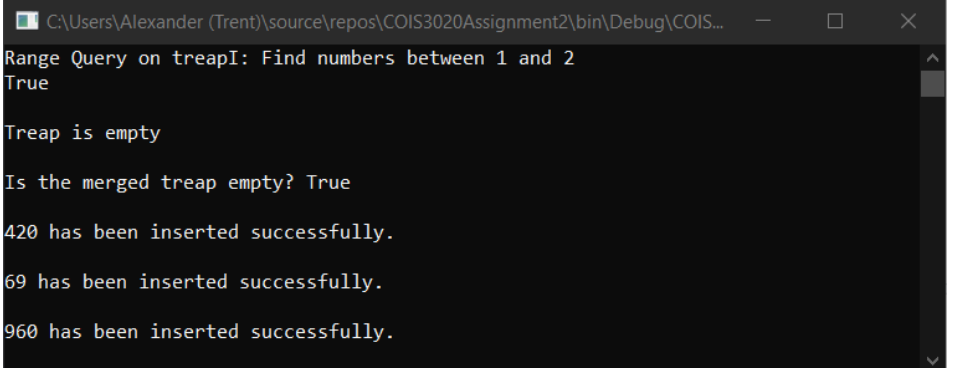
Case 5: Splitting an Empty Treap	
Description	This case is to test if splitting an Empty Treap fails immediately.
Expected Output	Treap is empty.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... If the code made it here, then that means that the Int Treap has been created. If the code made it here, then that means that the Char Treap has been created. Could not delete the item 1 because there is no root. Is 1 in treapI? False Range Query on treapI: Find numbers between 1 and 2 True Treap is empty </pre>

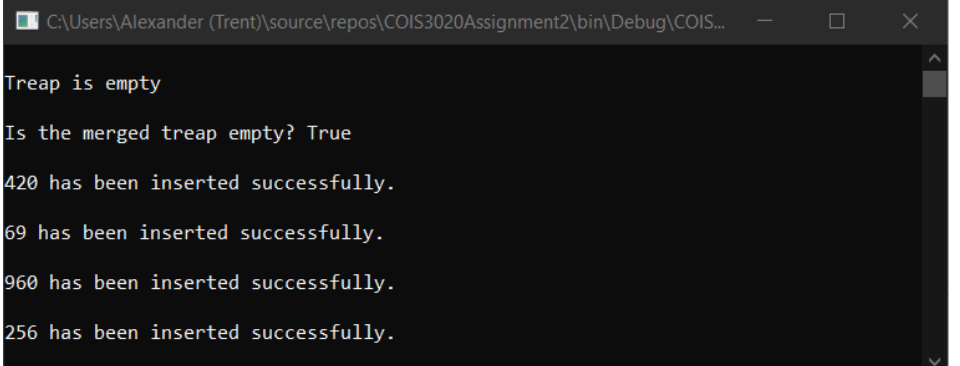
Case 6: Merging Two Empty Treaps of the Same Type	
Description	This case is to test if merging two empty treaps of the same types returns an empty treap.
Expected Output	Is the merged treap empty? True
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... If the code made it here, then that means that the Char Treap has been created. Could not delete the item 1 because there is no root. Is 1 in treapI? False Range Query on treapI: Find numbers between 1 and 2 True Treap is empty Is the merged treap empty? True </pre>

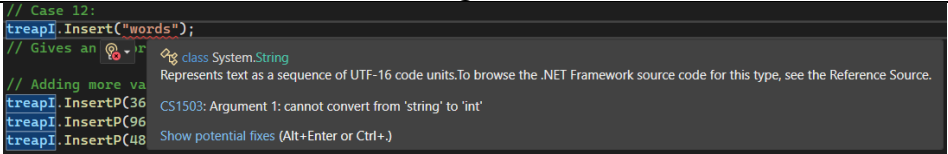
Case 7: Merging Two Empty Treaps of Different Types	
Description	This case is to test if merging two empty treaps of different types fails before running the code itself.
Expected Output	CS1503: cannot convert from COIS3020Assignment2.Treap<char> to COIS3020Assignment2.Treap<int>
Actual Output	 <pre> // Case 7: treap<int> treapIV = treapOne.Merge(treapI, treapC); // Gives an error before running, which is @.a (local variable) Treap<char> treapC // Cases 8-11 with fixed priorities treapI.InsertP(420, 70); Console.ReadLine(); </pre> <p>CS1503: Argument 2: cannot convert from 'COIS3020Assignment2.Treap<char>' to 'COIS3020Assignment2.Treap<int>'</p> <p>Show potential fixes (Alt+Enter or Ctrl+.)</p>

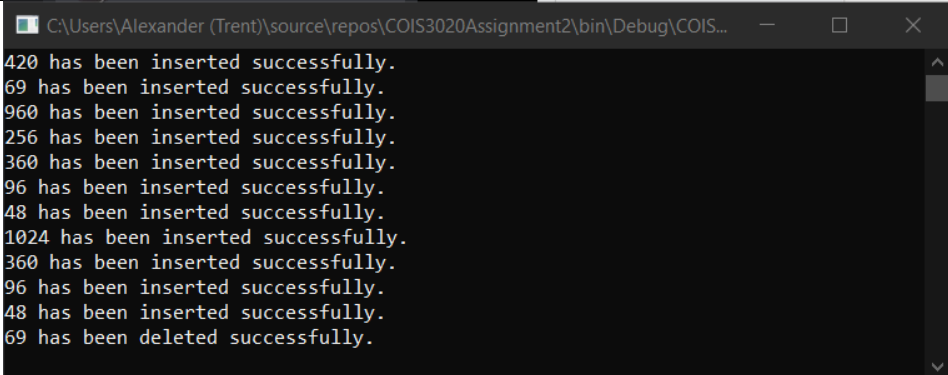
Case 8: Node Creation and Inserting into an Empty Treap	
Description	This case is to test if a node can be created and inserted into an empty treap without failure.
Expected Output	420 has been inserted successfully.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... Could not delete the item 1 because there is no root. Is 1 in treapI? False Range Query on treapI: Find numbers between 1 and 2 True Treap is empty Is the merged treap empty? True 420 has been inserted successfully. </pre>

Case 9: Inserting a Left Child into a Treap	
Description	This case is to test if a node can be inserted into a Treap as a left child without failure.
Expected Output	69 has been inserted successfully.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... Is 1 in treapI? False Range Query on treapI: Find numbers between 1 and 2 True Treap is empty Is the merged treap empty? True 420 has been inserted successfully. 69 has been inserted successfully. </pre>

Case 10: Inserting a Right Child into a Treap	
Description	This case is to test if a node can be inserted into a Treap as a right child without failure.
Expected Output	960 has been inserted successfully.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... Range Query on treapI: Find numbers between 1 and 2 True Treap is empty Is the merged treap empty? True 420 has been inserted successfully. 69 has been inserted successfully. 960 has been inserted successfully. </pre>

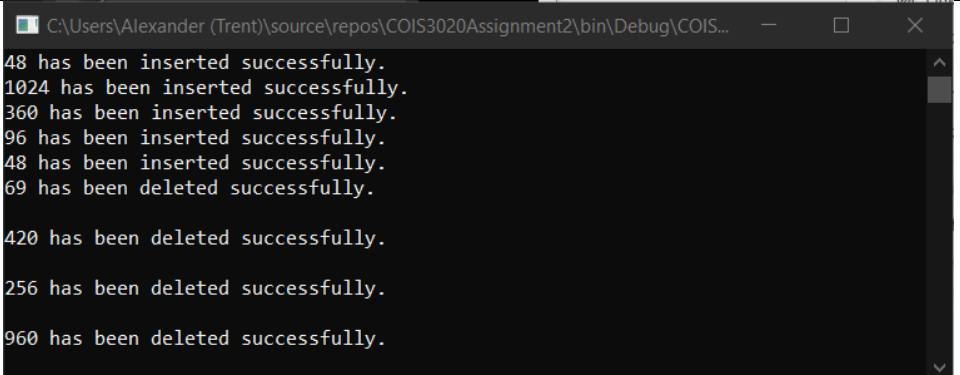
Case 11: Inserting a New Root into a Treap	
Description	This case is to test if a node can be inserted into a Treap as a new root without failure.
Expected Output	256 has been inserted successfully.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... Treap is empty Is the merged treap empty? True 420 has been inserted successfully. 69 has been inserted successfully. 960 has been inserted successfully. 256 has been inserted successfully. </pre>

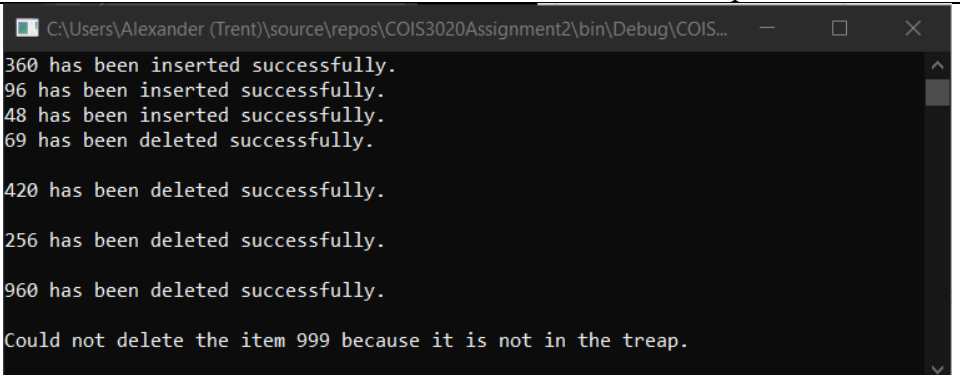
Case 12: Inserting an Invalid Node into a Treap	
Description	This case is to test if an invalid node can not be inserted into a Treap without throwing errors during compiling.
Expected Output	CS1503: cannot convert from 'string' to 'int'
Actual Output	<pre>// Case 12: treapI.Insert("words"); // Gives an error // Adding more va treapI.InsertP(36 treapI.InsertP(96 treapI.InsertP(48</pre> 

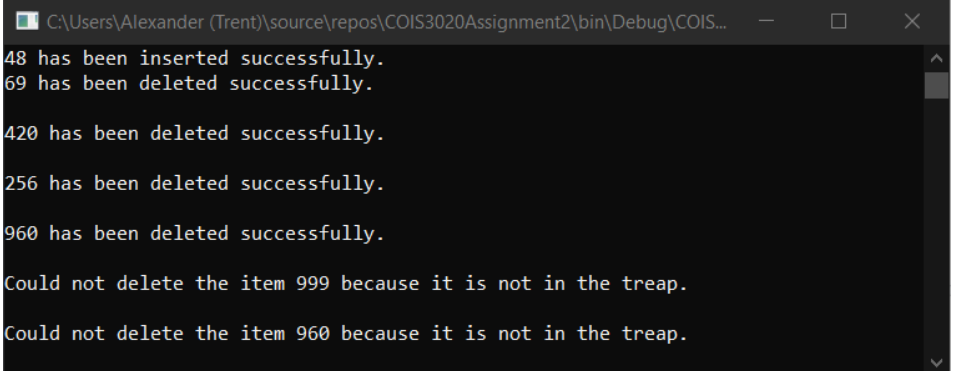
Case 13: Removing a Left Child in a Treap	
Description	This case is to test if a left child node can be removed in a Treap without failure.
Expected Output	// After many insertions (both with manual and random priorities) 69 has been deleted successfully.
Actual Output	 <pre>C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 420 has been inserted successfully. 69 has been inserted successfully. 960 has been inserted successfully. 256 has been inserted successfully. 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 1024 has been inserted successfully. 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 69 has been deleted successfully.</pre>

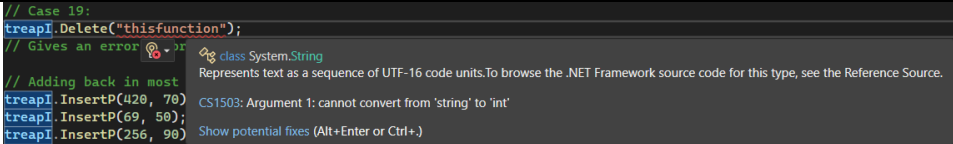
Case 14: Removing a Right Child in a Treap	
Description	This case is to test if a right child node can be removed in a Treap without failure.
Expected Output	420 has been deleted successfully.
Actual Output	 <pre>C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 960 has been inserted successfully. 256 has been inserted successfully. 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 1024 has been inserted successfully. 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 69 has been deleted successfully. 420 has been deleted successfully.</pre>

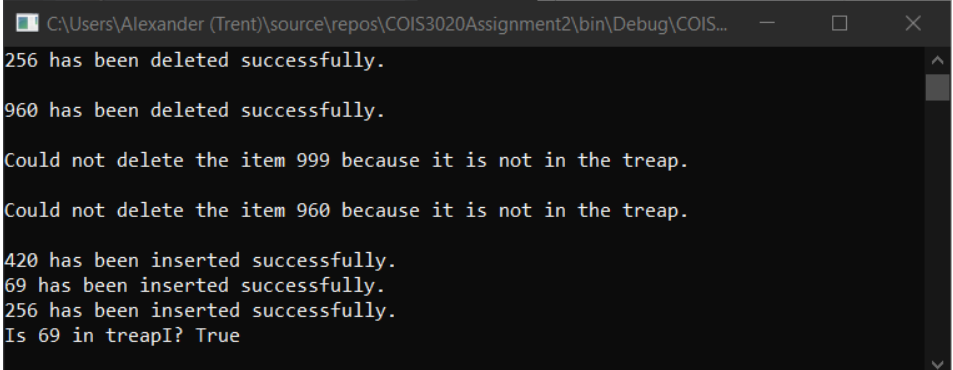
Case 15: Removing the Root in a Treap	
Description	This case is to test if the root node can be removed in a Treap without failure.
Expected Output	256 has been deleted successfully.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 1024 has been inserted successfully. 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 69 has been deleted successfully. 420 has been deleted successfully. 256 has been deleted successfully. </pre>

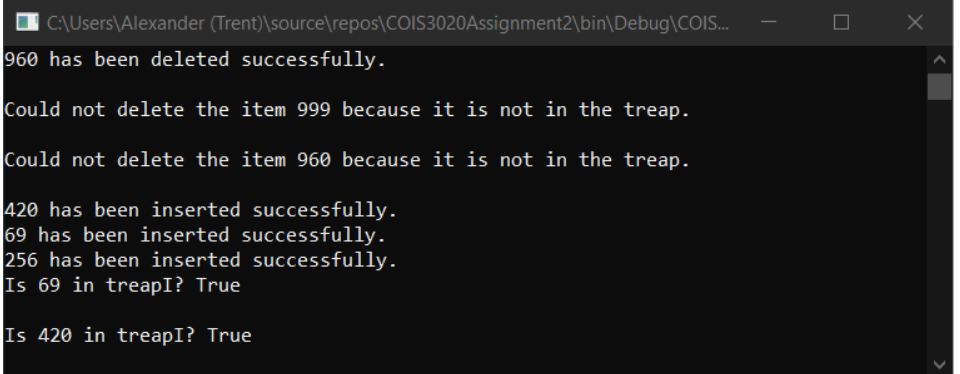
Case 16: Removing a Leaf in a Treap	
Description	This case is to test if a leaf node can be removed in a Treap without failure.
Expected Output	960 has been deleted successfully.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 48 has been inserted successfully. 1024 has been inserted successfully. 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 69 has been deleted successfully. 420 has been deleted successfully. 256 has been deleted successfully. 960 has been deleted successfully. </pre>

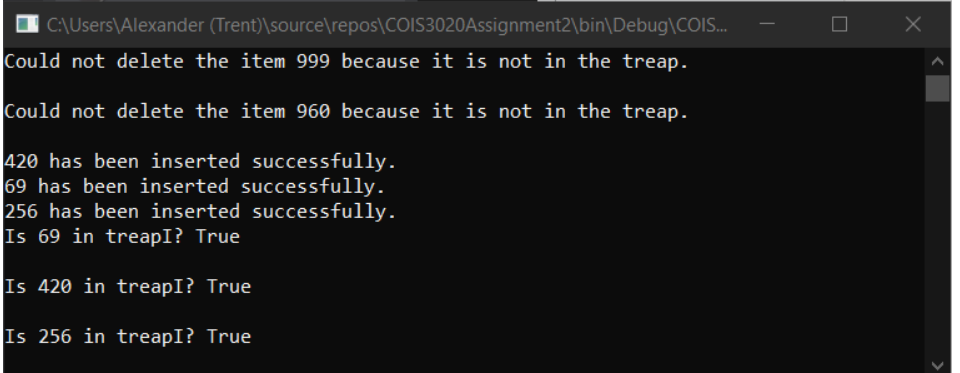
Case 17: Removing a Node that Doesn't Exist in a Treap	
Description	This case is to test if removing a node that doesn't exist in a Treap fails.
Expected Output	Could not delete the item 999 because it is not in the treap.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 360 has been inserted successfully. 96 has been inserted successfully. 48 has been inserted successfully. 69 has been deleted successfully. 420 has been deleted successfully. 256 has been deleted successfully. 960 has been deleted successfully. Could not delete the item 999 because it is not in the treap. </pre>

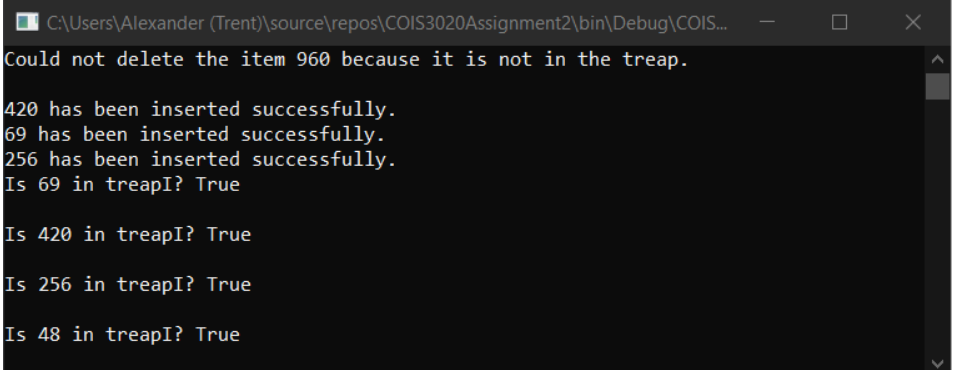
Case 18: Removing a Node that has Already Been Deleted in a Treap	
Description	This case is to test if removing a node that has already been deleted in a Treap fails.
Expected Output	Could not delete the item 960 because it is not in the treap.
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 48 has been inserted successfully. 69 has been deleted successfully. 420 has been deleted successfully. 256 has been deleted successfully. 960 has been deleted successfully. Could not delete the item 999 because it is not in the treap. Could not delete the item 960 because it is not in the treap. </pre>

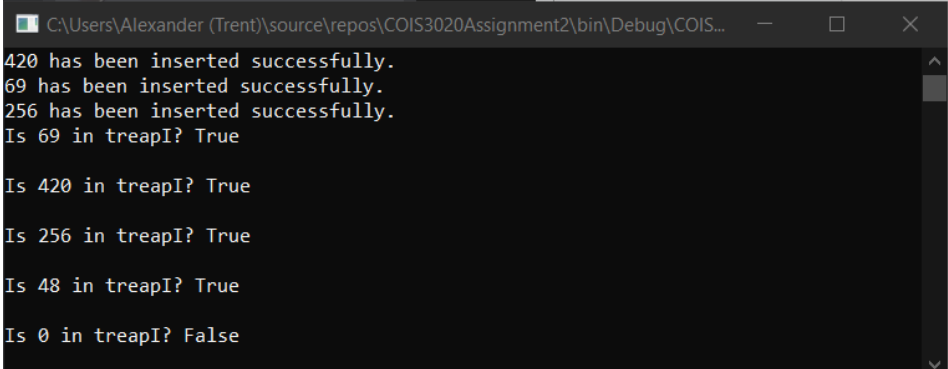
Case 19: Removing an Invalid Node in a Treap	
Description	This case is to test if removing an invalid node in a Treap can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from 'string' to 'int'
Actual Output	 <pre> // Case 19: treapI.Delete(thisfunction); // Gives an error // Adding back in most treapI.InsertP(420, 70); treapI.InsertP(69, 50); treapI.InsertP(256, 90); </pre>

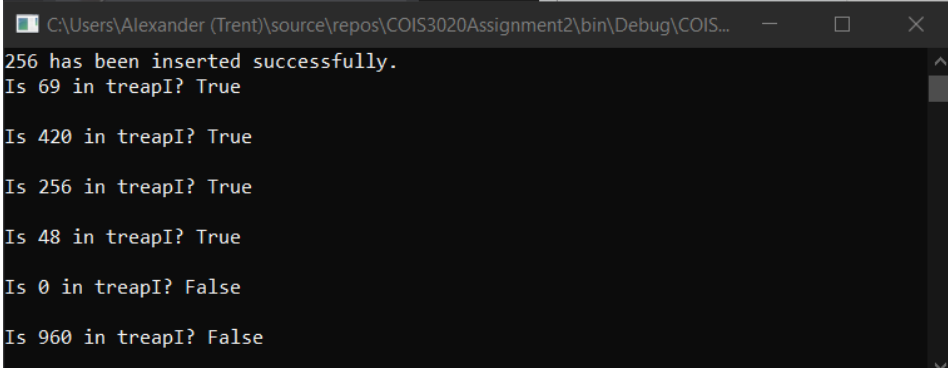
Case 20: Searching for a Left Child in a Treap	
Description	This case is to test if a left child node can be found in a Treap without failure.
Expected Output	// After some more insertions Is 69 in treapI? True
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 256 has been deleted successfully. 960 has been deleted successfully. Could not delete the item 999 because it is not in the treap. Could not delete the item 960 because it is not in the treap. 420 has been inserted successfully. 69 has been inserted successfully. 256 has been inserted successfully. Is 69 in treapI? True </pre>

Case 21: Searching for a Right Child in a Treap	
Description	This case is to test if a right child node can be found in a Treap without failure.
Expected Output	Is 420 in treapI? True
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 960 has been deleted successfully. Could not delete the item 999 because it is not in the treap. Could not delete the item 960 because it is not in the treap. 420 has been inserted successfully. 69 has been inserted successfully. 256 has been inserted successfully. Is 69 in treapI? True Is 420 in treapI? True </pre>

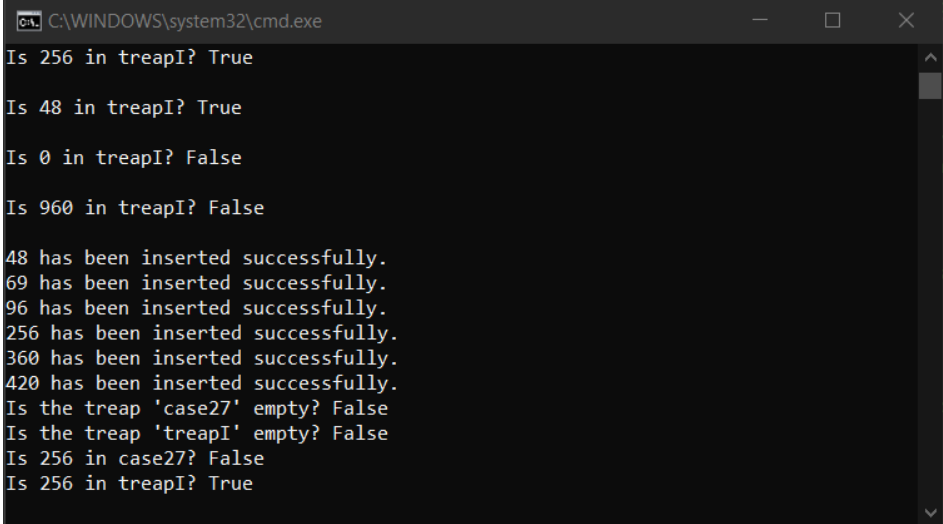
Case 22: Searching for the Root in a Treap	
Description	This case is to test if the root node can be found in a Treap without failure.
Expected Output	Is 256 in treapI? True
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... Could not delete the item 999 because it is not in the treap. Could not delete the item 960 because it is not in the treap. 420 has been inserted successfully. 69 has been inserted successfully. 256 has been inserted successfully. Is 69 in treapI? True Is 420 in treapI? True Is 256 in treapI? True </pre>

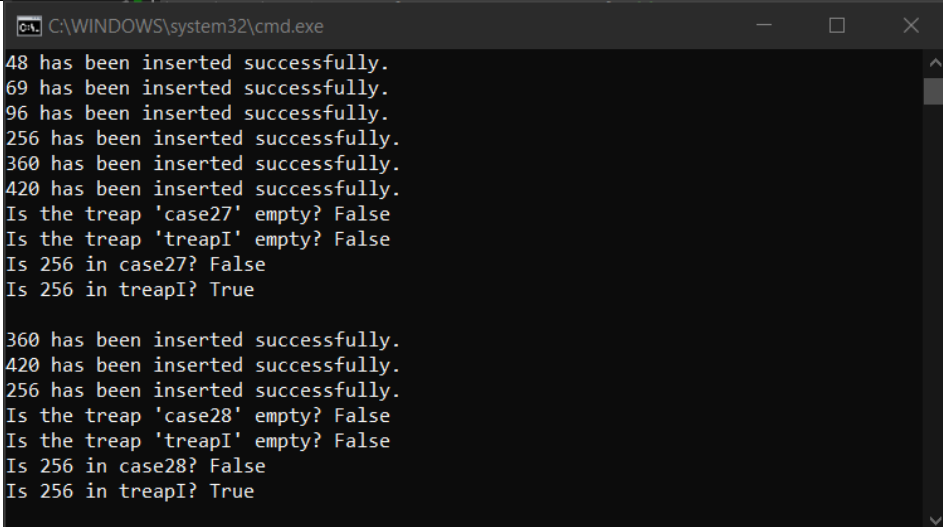
Case 23: Searching for a Leaf in a Treap	
Description	This case is to test if a leaf node can be found in a Treap without failure.
Expected Output	Is 48 in treapI? True
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... Could not delete the item 960 because it is not in the treap. 420 has been inserted successfully. 69 has been inserted successfully. 256 has been inserted successfully. Is 69 in treapI? True Is 420 in treapI? True Is 256 in treapI? True Is 48 in treapI? True </pre>

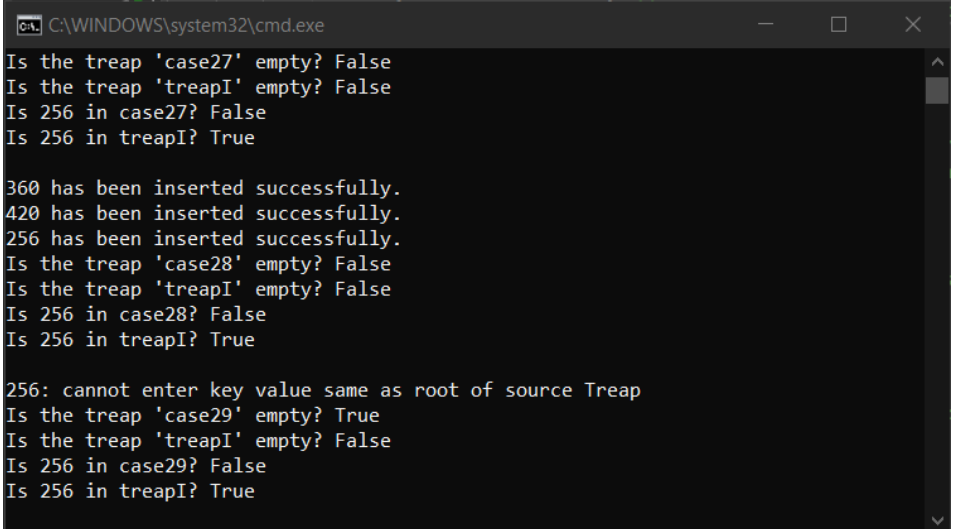
Case 24: Searching for a Node that Doesn't Exist in a Treap	
Description	This case is to test if finding a node that doesn't exist in a Treap returns null.
Expected Output	Is 0 in treapI? False
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 420 has been inserted successfully. 69 has been inserted successfully. 256 has been inserted successfully. Is 69 in treapI? True Is 420 in treapI? True Is 256 in treapI? True Is 48 in treapI? True Is 0 in treapI? False </pre>

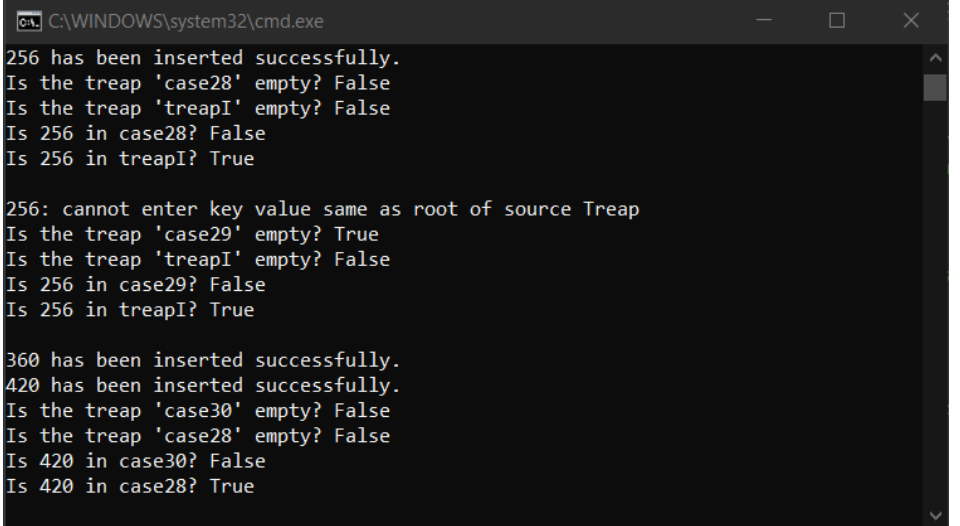
Case 25: Searching for a Node that has Already Been Deleted in a Treap	
Description	This case is to test if finding a node that has already been deleted in a Treap returns null.
Expected Output	Is 960 in treapI? False
Actual Output	 <pre> C:\Users\Alexander (Trent)\source\repos\COIS3020Assignment2\bin\Debug\COIS... 256 has been inserted successfully. Is 69 in treapI? True Is 420 in treapI? True Is 256 in treapI? True Is 48 in treapI? True Is 0 in treapI? False Is 960 in treapI? False </pre>

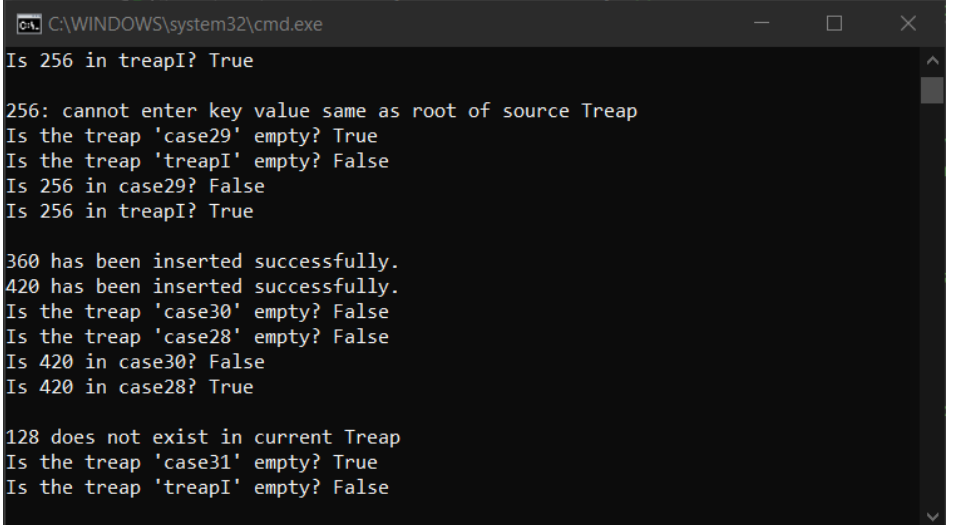
Case 26: Searching for an Invalid Node in a Treap	
Description	This case is to test if finding an invalid node in a Treap can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from 'string' to 'int'
Actual Output	 <pre> // Case 26: Console.WriteLine("Is words in treapI? " + treapI.Search("forthe7seas")); // Gives an error before running, which is what it should. // Case 27-32 treap<int> case27 = treapI.Split(69); Console.WriteLine("Is the treap \"case27\" empty? " + case27.IsEmpty()); Console.WriteLine("Is the treap \"treapI\" empty? " + treapI.IsEmpty()); </pre>

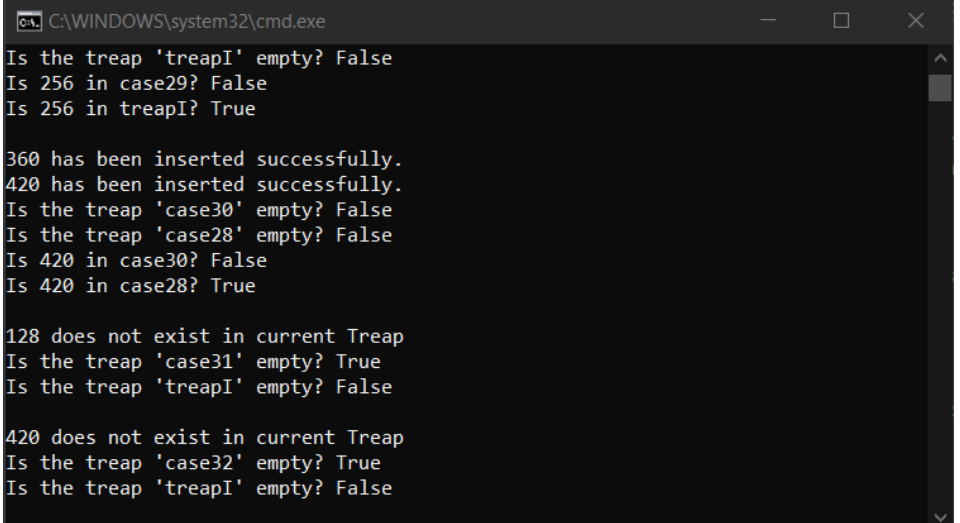
Case 27: Splitting a Treap at a Left Child	
Description	This case is to test if splitting a Treap at a left child produces two valid treaps.
Expected Output	// Insertions based on InsertP Is the treap 'case27' empty? False Is the treap 'treapI' empty? False Is 256 in case27? False Is 256 in treapI? True
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe Is 256 in treapI? True Is 48 in treapI? True Is 0 in treapI? False Is 960 in treapI? False 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 256 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case27' empty? False Is the treap 'treapI' empty? False Is 256 in case27? False Is 256 in treapI? True </pre>

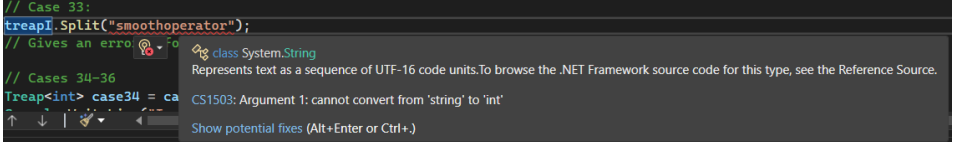
Case 28: Splitting a Treap at a Right Child	
Description	This case is to test if splitting a Treap at a right child produces two valid treaps.
Expected Output	// Insertions based on InsertP Is the treap 'case28' empty? False Is the treap 'treapI' empty? False Is 256 in case28? False Is 256 in treapI? True
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 256 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case27' empty? False Is the treap 'treapI' empty? False Is 256 in case27? False Is 256 in treapI? True 360 has been inserted successfully. 420 has been inserted successfully. 256 has been inserted successfully. Is the treap 'case28' empty? False Is the treap 'treapI' empty? False Is 256 in case28? False Is 256 in treapI? True </pre>

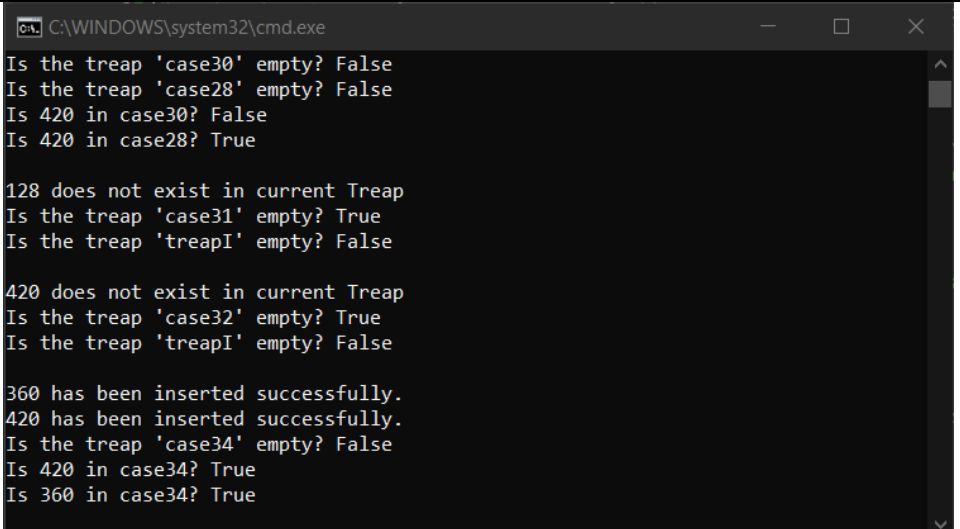
Case 29: Splitting a Treap at the Root	
Description	This case is to test if splitting a Treap at the root fails.
Expected Output	256: cannot enter key value same as root of source Treap Is the treap 'case29' empty? True Is the treap 'treapI' empty? False Is 256 in case29? False Is 256 in treapI? True
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe Is the treap 'case27' empty? False Is the treap 'treapI' empty? False Is 256 in case27? False Is 256 in treapI? True 360 has been inserted successfully. 420 has been inserted successfully. 256 has been inserted successfully. Is the treap 'case28' empty? False Is the treap 'treapI' empty? False Is 256 in case28? False Is 256 in treapI? True 256: cannot enter key value same as root of source Treap Is the treap 'case29' empty? True Is the treap 'treapI' empty? False Is 256 in case29? False Is 256 in treapI? True </pre>

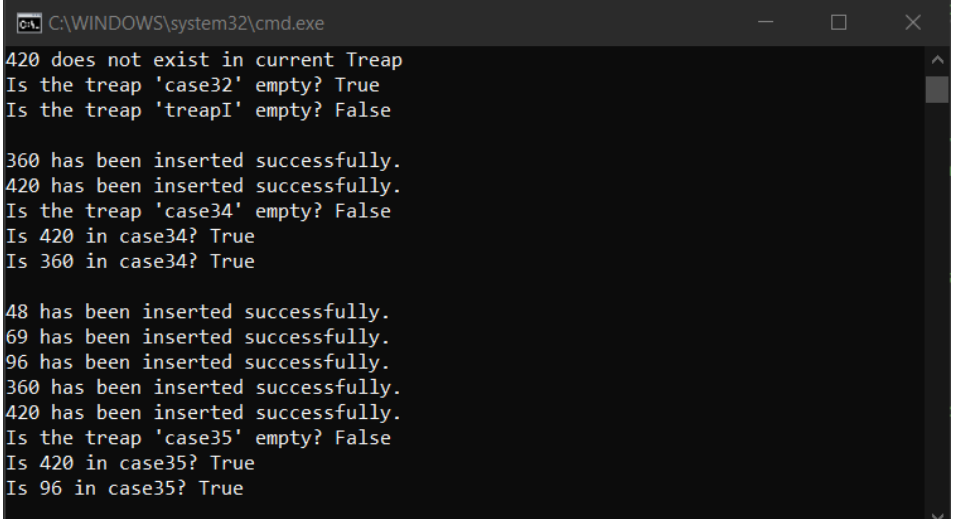
Case 30: Splitting a Treap at a Leaf	
Description	This case is to test if splitting a Treap at a leaf node produces two valid treaps.
Expected Output	<pre>// Insertions based on InsertP Is the treap 'case30' empty? False Is the treap 'case28' empty? False Is 420 in case30? False Is 420 in case28? True</pre>
Actual Output	 <pre>C:\WINDOWS\system32\cmd.exe 256 has been inserted successfully. Is the treap 'case28' empty? False Is the treap 'treapI' empty? False Is 256 in case28? False Is 256 in treapI? True 256: cannot enter key value same as root of source Treap Is the treap 'case29' empty? True Is the treap 'treapI' empty? False Is 256 in case29? False Is 256 in treapI? True 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case30' empty? False Is the treap 'case28' empty? False Is 420 in case30? False Is 420 in case28? True</pre>

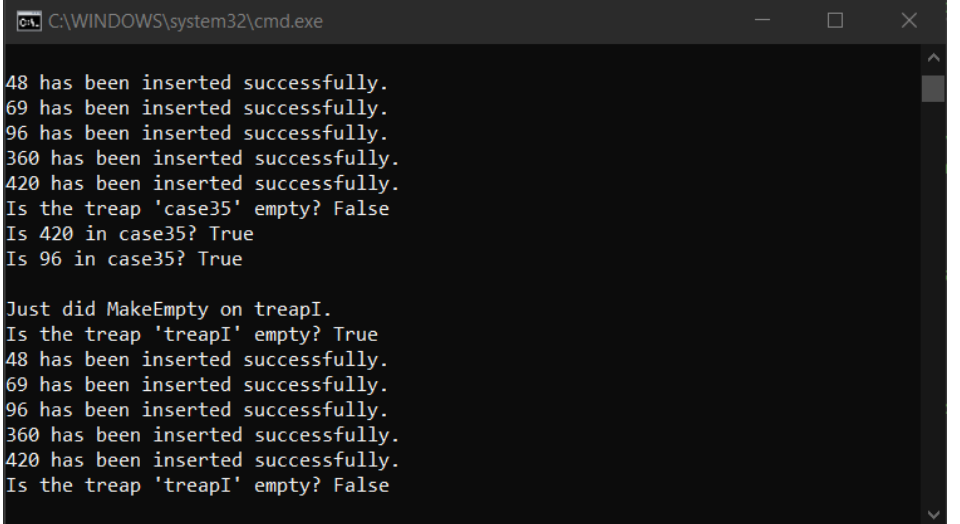
Case 31: Splitting a Treap at a Node that Doesn't Exist	
Description	This case is to test if splitting a Treap at a node that is non-existent fails.
Expected Output	<pre>128 does not exist in current Treap Is the treap 'case31' empty? True Is the treap 'treapI' empty? False</pre>
Actual Output	 <pre>C:\WINDOWS\system32\cmd.exe Is 256 in treapI? True 256: cannot enter key value same as root of source Treap Is the treap 'case29' empty? True Is the treap 'treapI' empty? False Is 256 in case29? False Is 256 in treapI? True 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case30' empty? False Is the treap 'case28' empty? False Is 420 in case30? False Is 420 in case28? True 128 does not exist in current Treap Is the treap 'case31' empty? True Is the treap 'treapI' empty? False</pre>

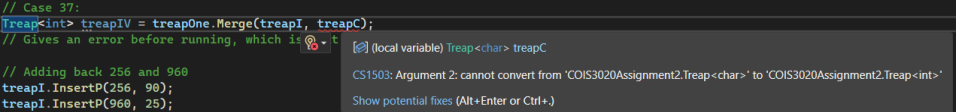
Case 32: Splitting a Treap at a Node that has Already Been Deleted	
Description	This case is to test if splitting a Treap at a node that has been inserted and then deleted fails.
Expected Output	420 does not exist in current Treap Is the treap 'case32' empty? True Is the treap 'treapI' empty? False
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe Is the treap 'treapI' empty? False Is 256 in case29? False Is 256 in treapI? True 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case30' empty? False Is the treap 'case28' empty? False Is 420 in case30? False Is 420 in case28? True 128 does not exist in current Treap Is the treap 'case31' empty? True Is the treap 'treapI' empty? False 420 does not exist in current Treap Is the treap 'case32' empty? True Is the treap 'treapI' empty? False </pre>

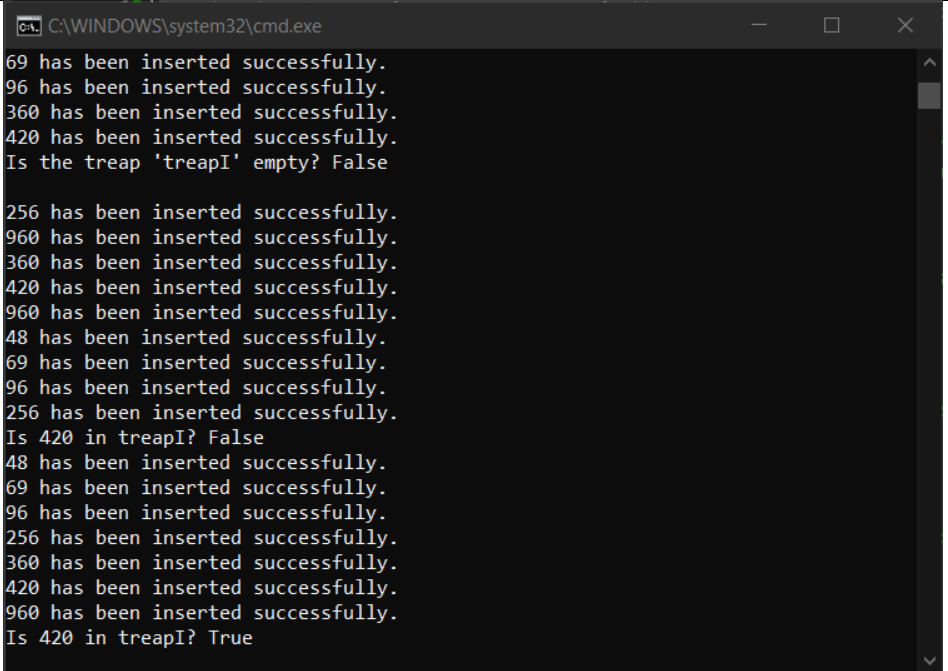
Case 33: Splitting a Treap at an Invalid Node	
Description	This case is to test if splitting an invalid node in a Treap can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from 'string' to 'int'
Actual Output	 <pre> // Case 33: treapI.Split("smoothoperator"); // Gives an error // Cases 34-36 Treap<int> case34 = ca </pre> <p>CS1503: Argument 1: cannot convert from 'string' to 'int'</p> <p>Show potential fixes (Alt+Enter or Ctrl+.)</p>

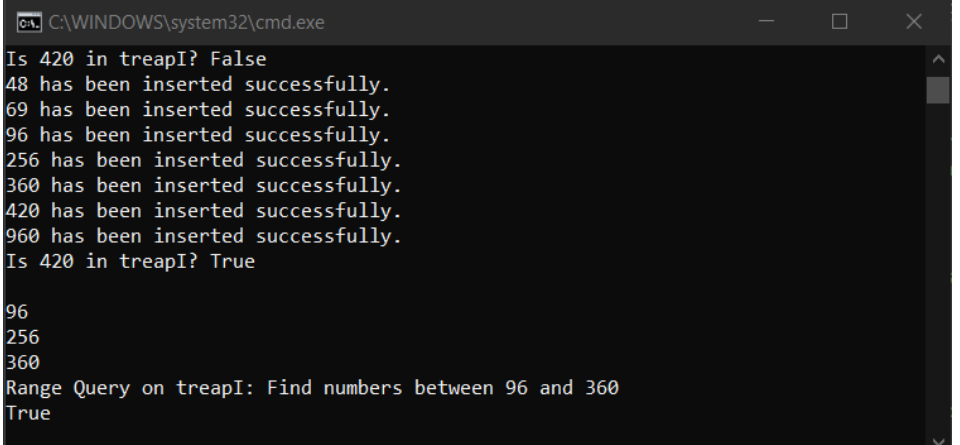
Case 34: Merging two Treaps with the Same Types and Balance	
Description	This case is to test if the merging of two treaps with the same type and balance make one valid treap.
Expected Output	// Insertions based on InsertP Is the treap 'case34' empty? False Is 420 in case34? True Is 360 in case34? True
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe Is the treap 'case30' empty? False Is the treap 'case28' empty? False Is 420 in case30? False Is 420 in case28? True 128 does not exist in current Treap Is the treap 'case31' empty? True Is the treap 'treapI' empty? False 420 does not exist in current Treap Is the treap 'case32' empty? True Is the treap 'treapI' empty? False 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case34' empty? False Is 420 in case34? True Is 360 in case34? True </pre>

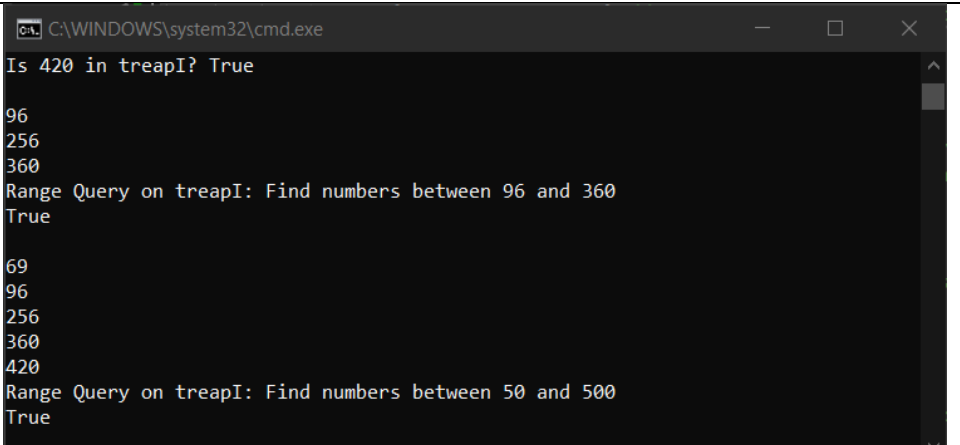
Case 35: Merging two Treaps with the Same Types, but with Different Balance	
Description	This case is to test if the merging of two treaps with the same type but with differing balance makes one valid treap.
Expected Output	// Insertions based on InsertP Is the treap 'case35' empty? False Is 420 in case35? True Is 96 in case35? True
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe 420 does not exist in current Treap Is the treap 'case32' empty? True Is the treap 'treapI' empty? False 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case34' empty? False Is 420 in case34? True Is 360 in case34? True 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case35' empty? False Is 420 in case35? True Is 96 in case35? True </pre>

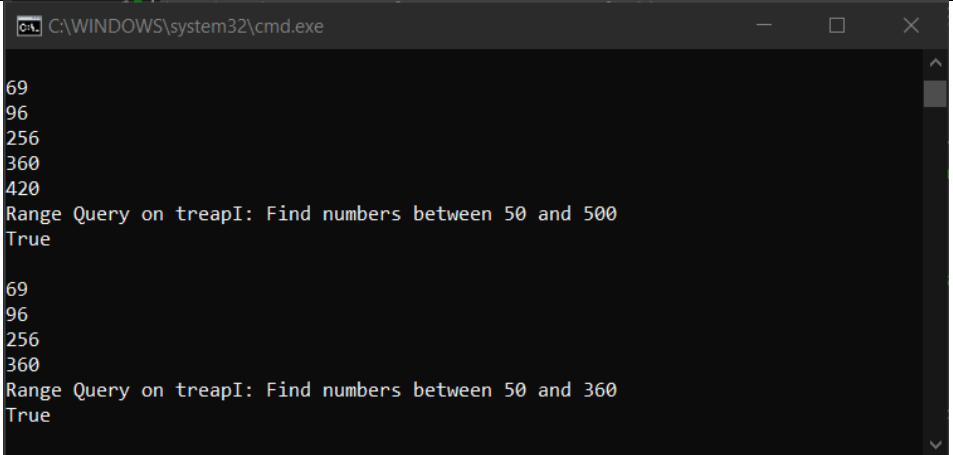
Case 36: Merging a Treap with an Empty Treap	
Description	This case is to test if the merging of a filled treap with an empty treap and the same type make one valid treap.
Expected Output	Just did MakeEmpty on treapI. Is the treap 'treapI' empty? True // After merge // Insertions based on InsertP Is the treap 'treapI' empty? False
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'case35' empty? False Is 420 in case35? True Is 96 in case35? True Just did MakeEmpty on treapI. Is the treap 'treapI' empty? True 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'treapI' empty? False </pre>

Case 37: Merging two Treaps with Different Types	
Description	This case is to test if the merging of two treaps with different types can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from COIS3020Assignment2.Treap<char> to COIS3020Assignment2.Treap<int>
Actual Output	 <pre> // Case 37: Treap<int> treapIV = treapOne.Merge(treapI, treapC); // Gives an error before running, which is // (local variable) Treap<char> treapC // Adding back 256 and 960 treapI.InsertP(256, 90); treapI.InsertP(960, 25); </pre> <p>CS1503: Argument 2: cannot convert from 'COIS3020Assignment2.Treap<char>' to 'COIS3020Assignment2.Treap<int>'</p> <p>Show potential fixes (Alt+Enter or Ctrl+.)</p>

Case 38: Splitting a Treap and then Merging it back Together	
Description	This case is to test if the splitting and merging of a treap returns the same treap that it started off at.
Expected Output	<pre>// After some more insertions // Split // Insertions based on InsertP Is 420 in treapI? False // Merge // Insertions based on InsertP Is 420 in treapI? True</pre>
Actual Output	 <pre>C:\WINDOWS\system32\cmd.exe 69 has been inserted successfully. 96 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. Is the treap 'treapI' empty? False 256 has been inserted successfully. 960 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. 960 has been inserted successfully. 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 256 has been inserted successfully. Is 420 in treapI? False 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 256 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. 960 has been inserted successfully. Is 420 in treapI? True</pre>

Case 39: Range Querying between two Nodes that Both Exist	
Description	This case is to test if the range querying between two nodes that both exist in a treap print a result that is correct with the query.
Expected Output	Range Query on treapI: Find numbers between 96 and 360 96 256 360 True
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe Is 420 in treapI? False 48 has been inserted successfully. 69 has been inserted successfully. 96 has been inserted successfully. 256 has been inserted successfully. 360 has been inserted successfully. 420 has been inserted successfully. 960 has been inserted successfully. Is 420 in treapI? True 96 256 360 Range Query on treapI: Find numbers between 96 and 360 True </pre>

Case 40: Range Querying between two Nodes that Both Don't Exist	
Description	This case is to test if the range querying between two nodes that both don't exist in a treap print a result that is correct with the query.
Expected Output	Range Query on treapI: Find numbers between 50 and 500 69 96 256 360 420 True
Actual Output	 <pre> C:\WINDOWS\system32\cmd.exe Is 420 in treapI? True 96 256 360 Range Query on treapI: Find numbers between 96 and 360 True 69 96 256 360 420 Range Query on treapI: Find numbers between 50 and 500 True </pre>

Case 41: Range Querying between a Node that Does Exist and Another that Doesn't	
Description	This case is to test if the range querying between two nodes, in which one exist in a treap and the other one doesn't prints a result that is correct with the query.
Expected Output	Range Query on treapI: Find numbers between 50 and 360 69 96 256 360 True
Actual Output	

Case 42: Range Querying with an Invalid Node in One Field	
Description	This case is to test if the range querying with an invalid node can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from 'string' to 'int'
Actual Output	