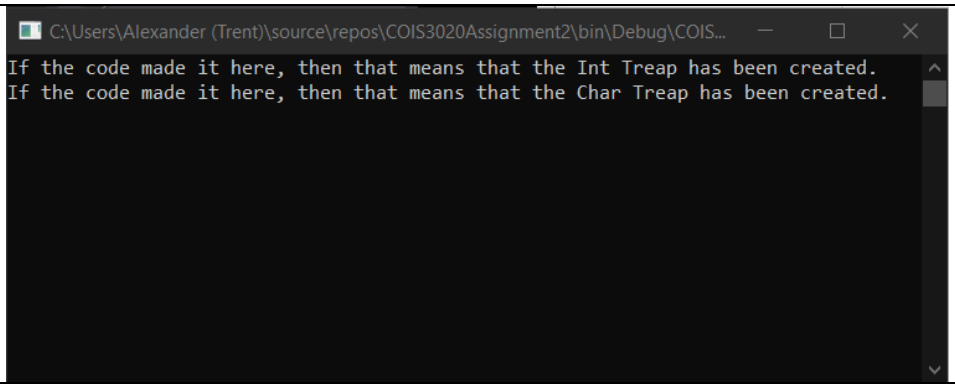
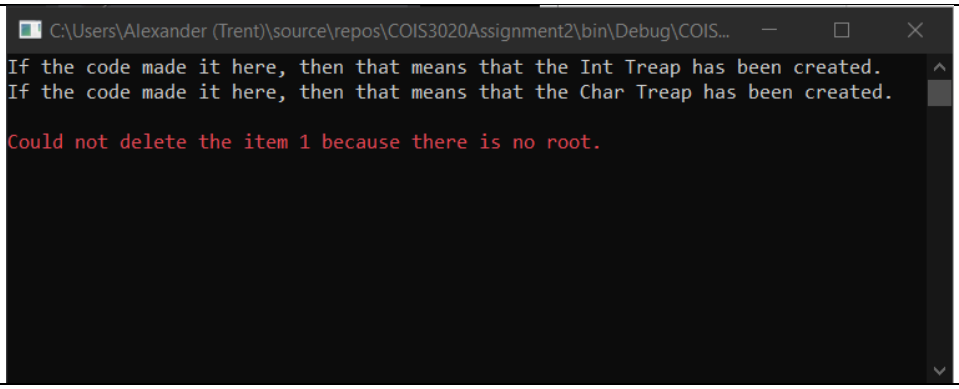
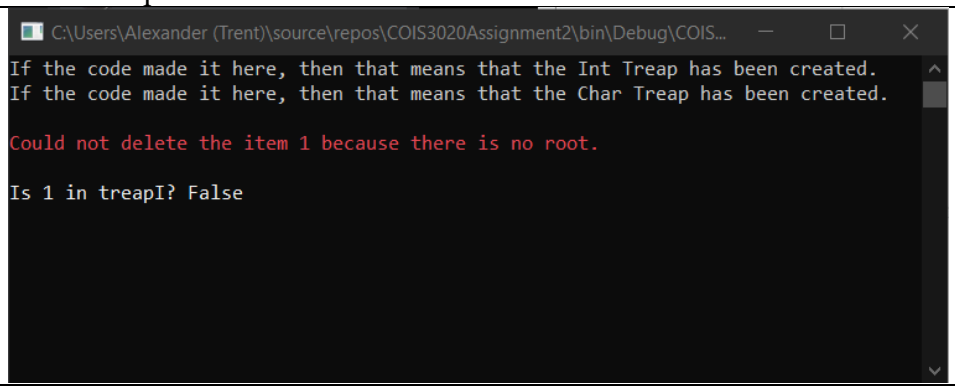
COIS 3020H: Data Structures and Algorithms II

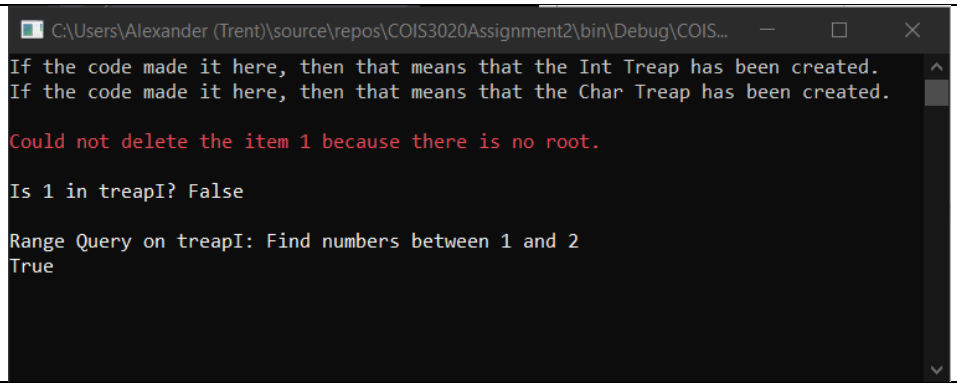
Assignment 2 Documentation and Analysis

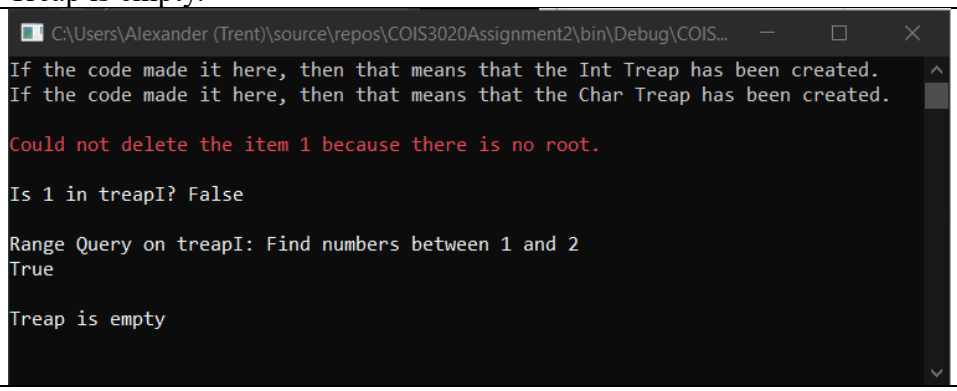
Due Date: November 20th, 2023

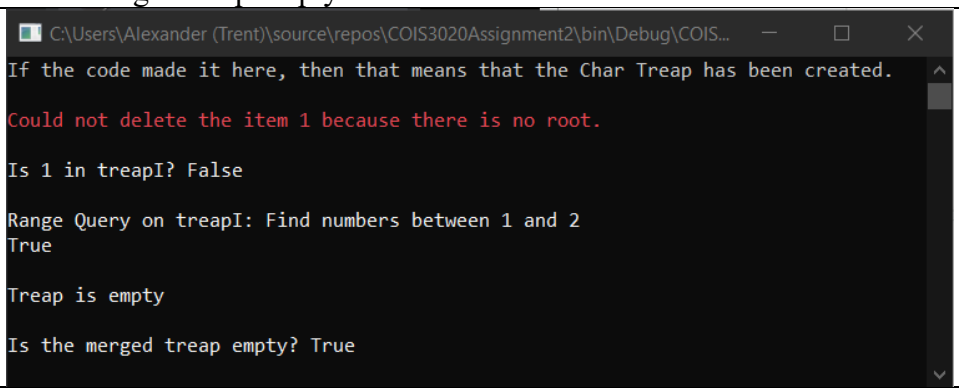
Case 1: Creating Treaps (Int and Char)	
Description	This case is to test if the Generic Treap Class can be created with multiple different data types without crashing.
Expected Output	If the code made it here, then that means that the Int Treap has been created. If the code made it here, then that means that the Char Treap has been created.
Actual Output	

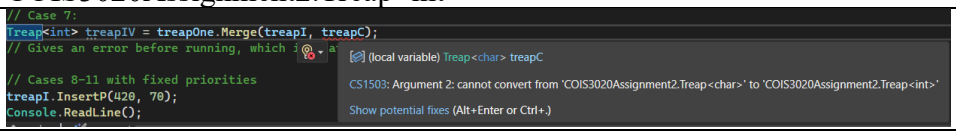
Case 2: Removing in an Empty Treap	
Description	This case is to test if removing in an Empty Treap fails immediately.
Expected Output	Could not delete the item 1 because there is no root.
Actual Output	

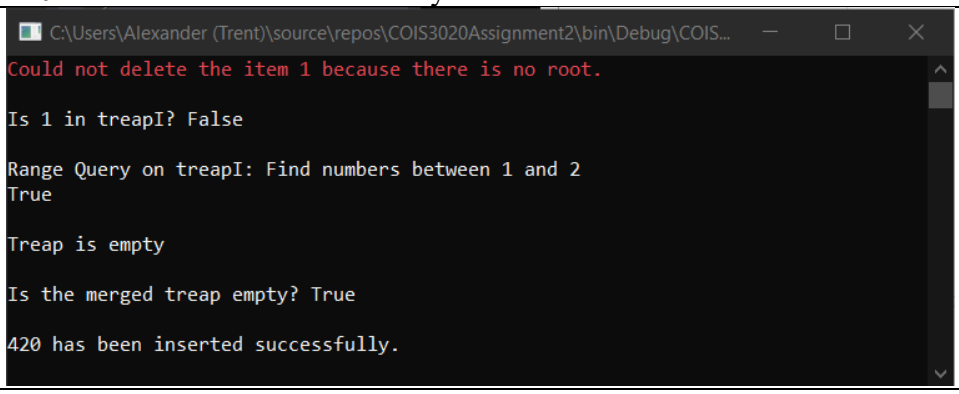
Case 3: Searching in an Empty Treap	
Description	This case is to test if searching in an Empty Treap returns false.
Expected Output	Is 1 in treapI? False
Actual Output	

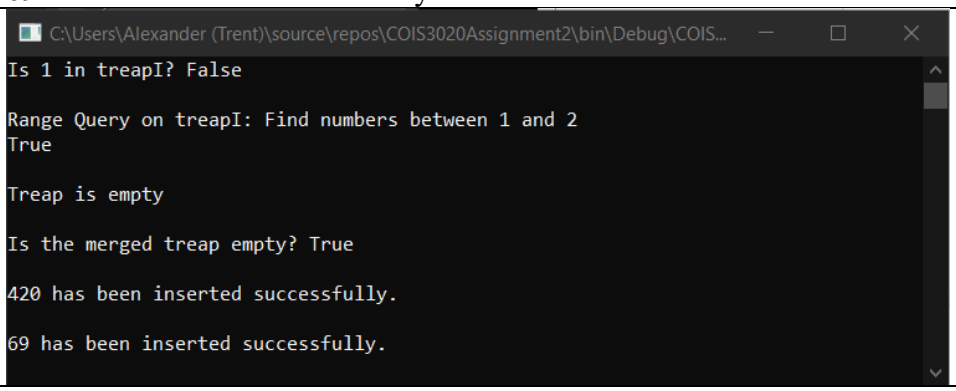
Case 4: Range Querying in an Empty Treap	
Description	This case is to test if range querying in an Empty Treap returns false.
Expected Output	Range Query on treapI: Find numbers between 1 and 2 True
Actual Output	

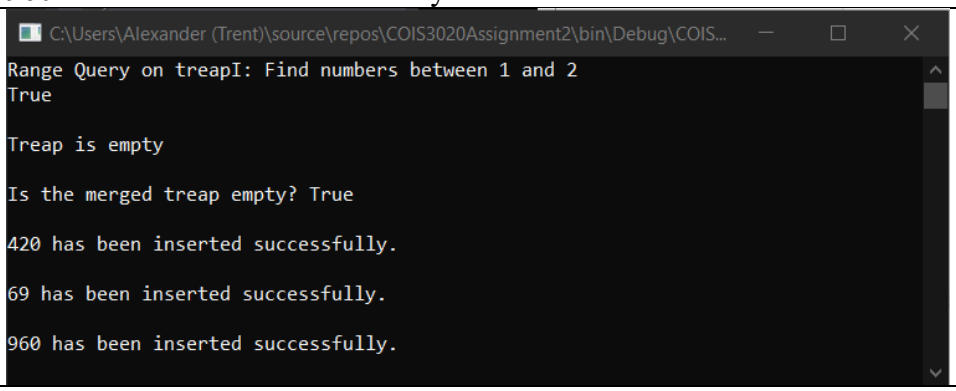
Case 5: Splitting an Empty Treap	
Description	This case is to test if splitting an Empty Treap fails immediately.
Expected Output	Treap is empty.
Actual Output	

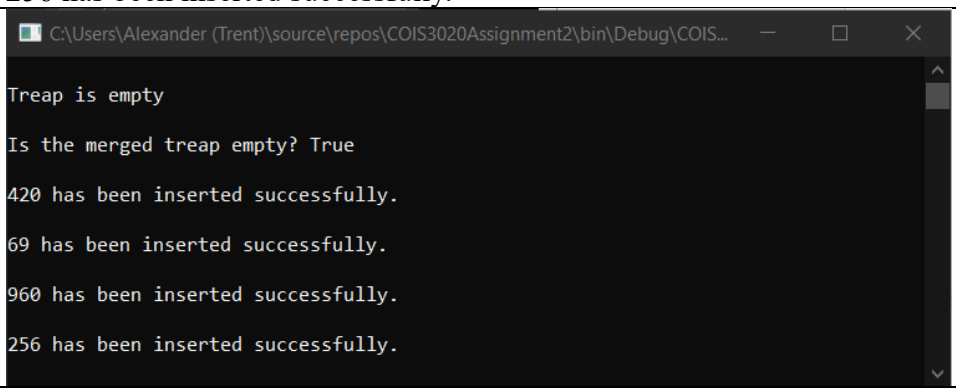
Case 6: Merging Two Empty Treaps of the Same Type	
Description	This case is to test if merging two empty treaps of the same types returns an empty treap.
Expected Output	Is the merged treap empty? True
Actual Output	

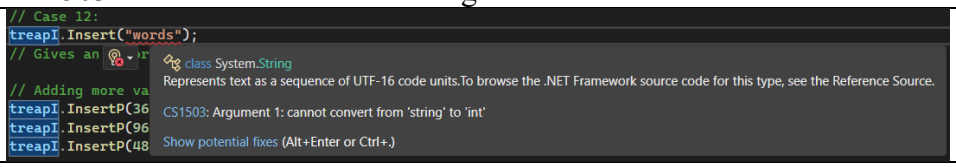
Case 7: Merging Two Empty Treaps of Different Types	
Description	This case is to test if merging two empty treaps of different types fails before running the code itself.
Expected Output	CS1503: cannot convert from COIS3020Assignment2.Treap<char> to COIS3020Assignment2.Treap<int>
Actual Output	

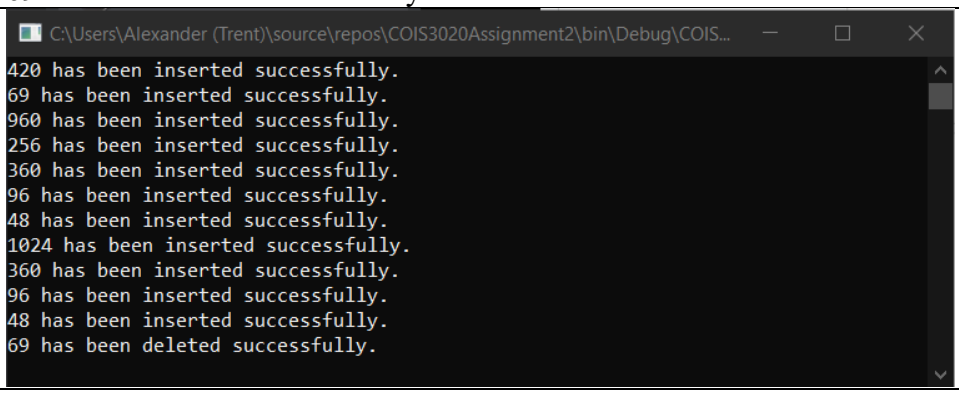
Case 8: Node Creation and Inserting into an Empty Treap	
Description	This case is to test if a node can be created and inserted into an empty treap without failure.
Expected Output	420 has been inserted successfully.
Actual Output	

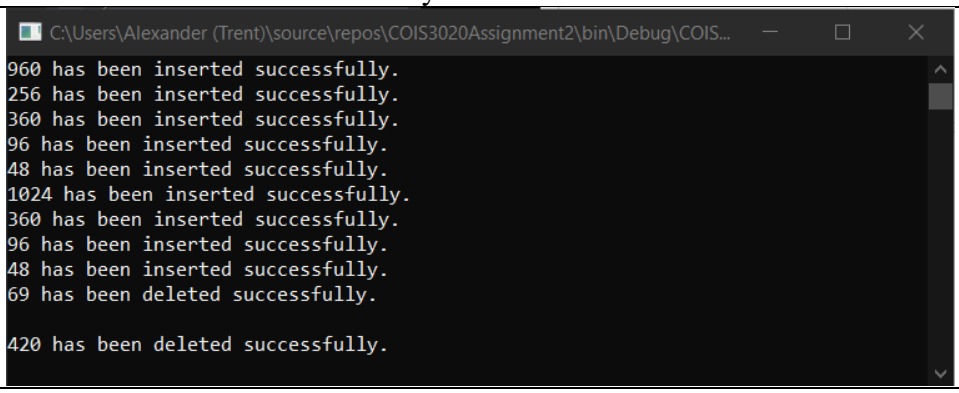
Case 9: Inserting a Left Child into a Treap	
Description	This case is to test if a node can be inserted into a Treap as a left child without failure.
Expected Output	69 has been inserted successfully.
Actual Output	

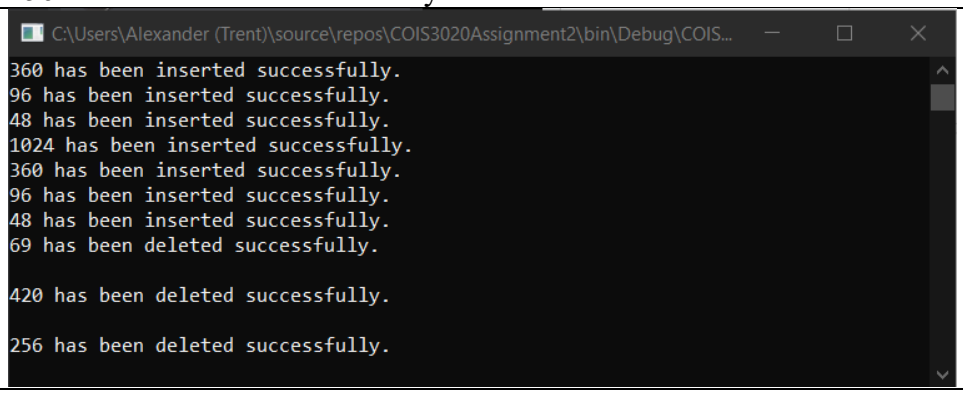
Case 10: Inserting a Right Child into a Treap	
Description	This case is to test if a node can be inserted into a Treap as a right child without failure.
Expected Output	960 has been inserted successfully.
Actual Output	

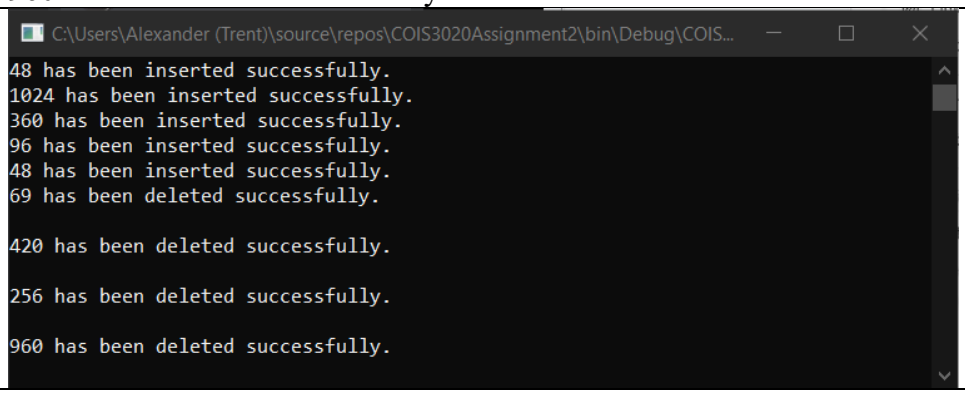
Case 11: Inserting a New Root into a Treap	
Description	This case is to test if a node can be inserted into a Treap as a new root without failure.
Expected Output	256 has been inserted successfully.
Actual Output	

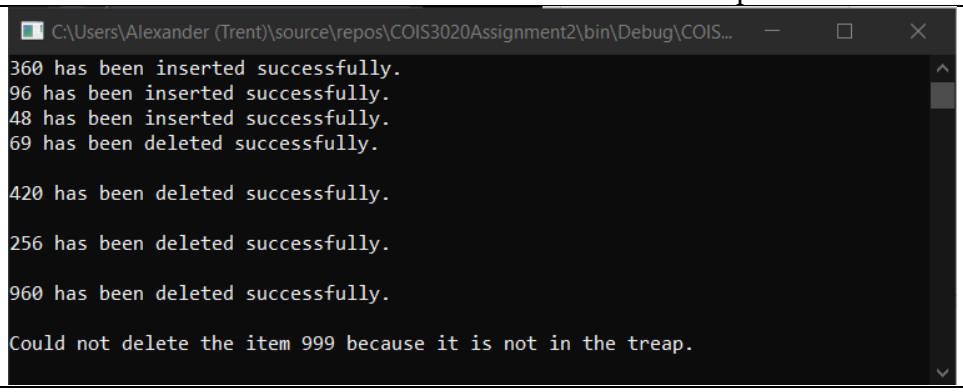
Case 12: Inserting an Invalid Node into a Treap	
Description	This case is to test if an invalid node can not be inserted into a Treap without throwing errors during compiling.
Expected Output	CS1503: cannot convert from ‘string’ to ‘int’
Actual Output	

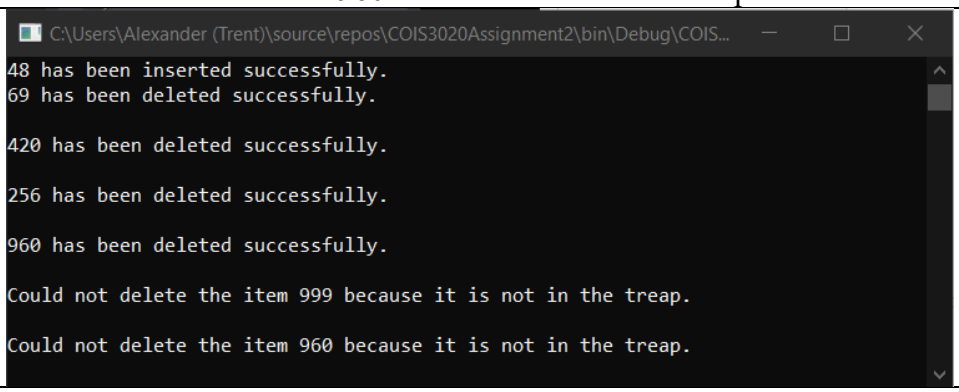
Case 13: Removing a Left Child in a Treap	
Description	This case is to test if a left child node can be removed in a Treap without failure.
Expected Output	// After many insertions (both with manual and random priorities) 69 has been deleted successfully.
Actual Output	

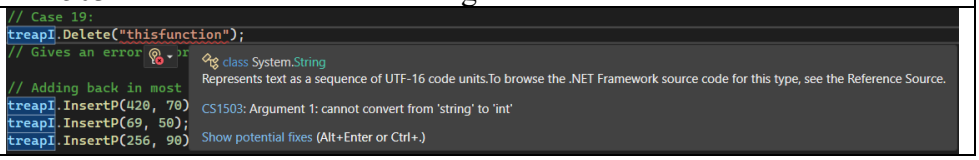
Case 14: Removing a Right Child in a Treap	
Description	This case is to test if a right child node can be removed in a Treap without failure.
Expected Output	420 has been deleted successfully.
Actual Output	

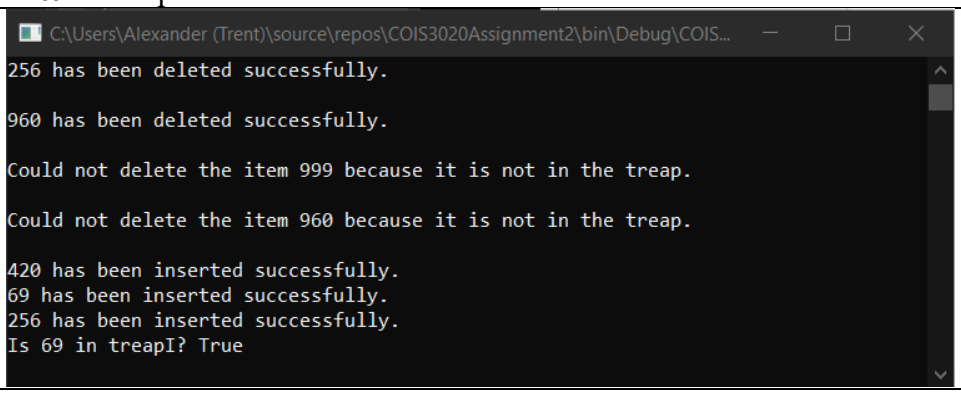
Case 15: Removing the Root in a Treap	
Description	This case is to test if the root node can be removed in a Treap without failure.
Expected Output	256 has been deleted successfully.
Actual Output	

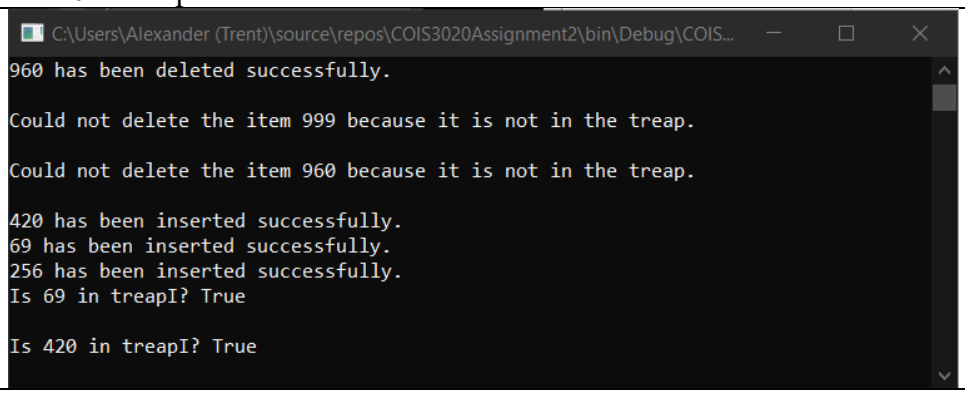
Case 16: Removing a Leaf in a Treap	
Description	This case is to test if a leaf node can be removed in a Treap without failure.
Expected Output	960 has been deleted successfully.
Actual Output	

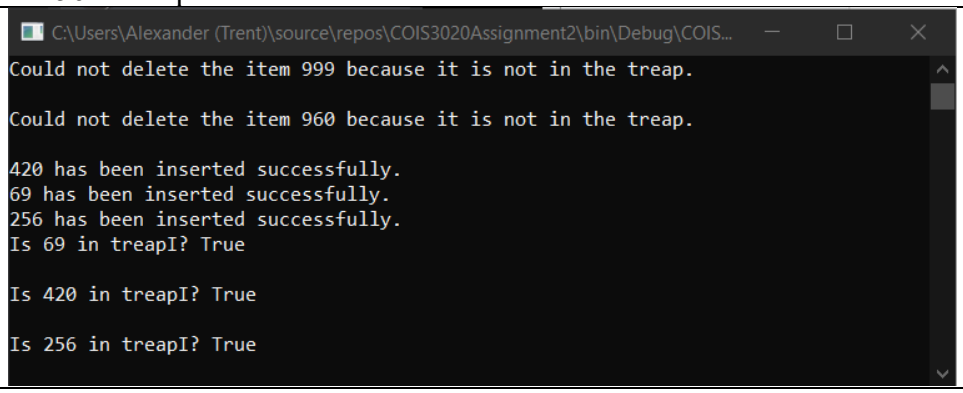
Case 17: Removing a Node that Doesn't Exist in a Treap	
Description	This case is to test if removing a node that doesn't exist in a Treap fails.
Expected Output	Could not delete the item 999 because it is not in the treap.
Actual Output	

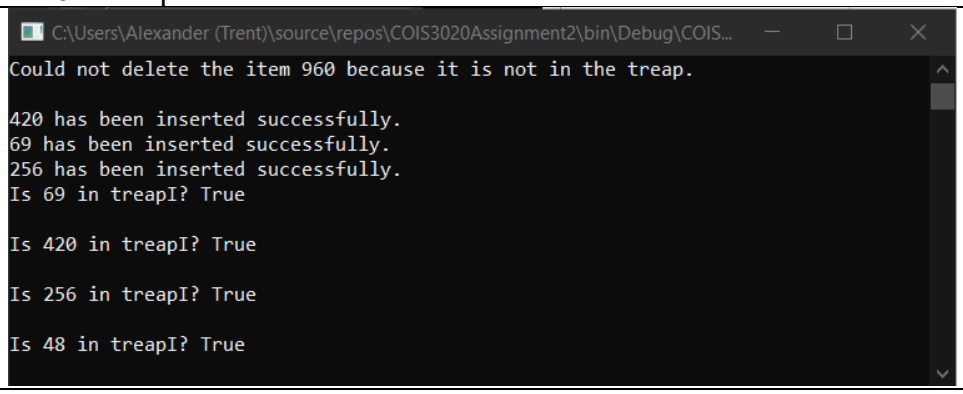
Case 18: Removing a Node that has Already Been Deleted in a Treap	
Description	This case is to test if removing a node that has already been deleted in a Treap fails.
Expected Output	Could not delete the item 960 because it is not in the treap.
Actual Output	

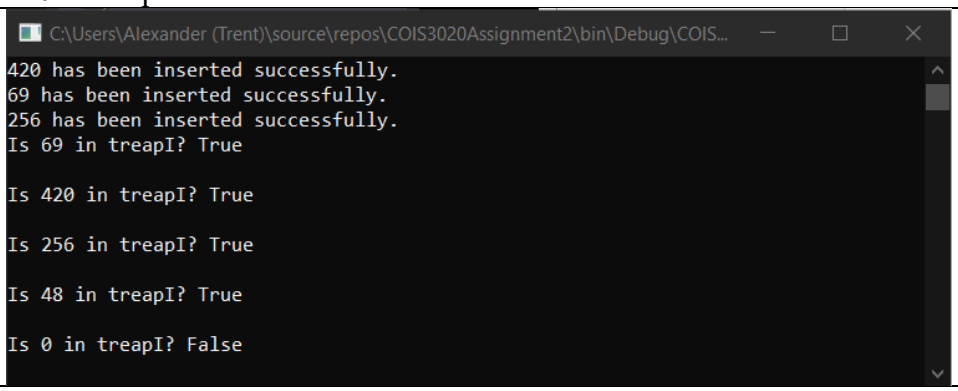
Case 19: Removing an Invalid Node in a Treap	
Description	This case is to test if removing an invalid node in a Treap can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from ‘string’ to ‘int’
Actual Output	

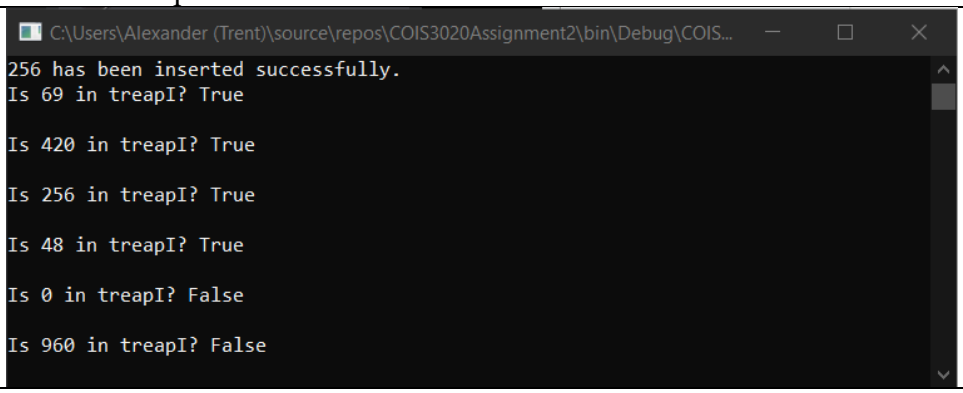
Case 20: Searching for a Left Child in a Treap	
Description	This case is to test if a left child node can be found in a Treap without failure.
Expected Output	// After some more insertions Is 69 in treapI? True
Actual Output	

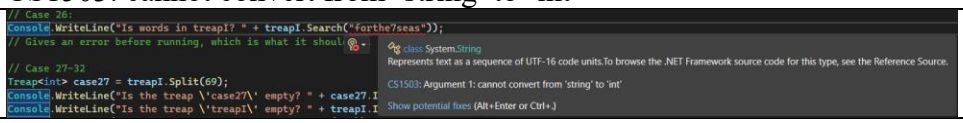
Case 21: Searching for a Right Child in a Treap	
Description	This case is to test if a right child node can be found in a Treap without failure.
Expected Output	Is 420 in treapI? True
Actual Output	

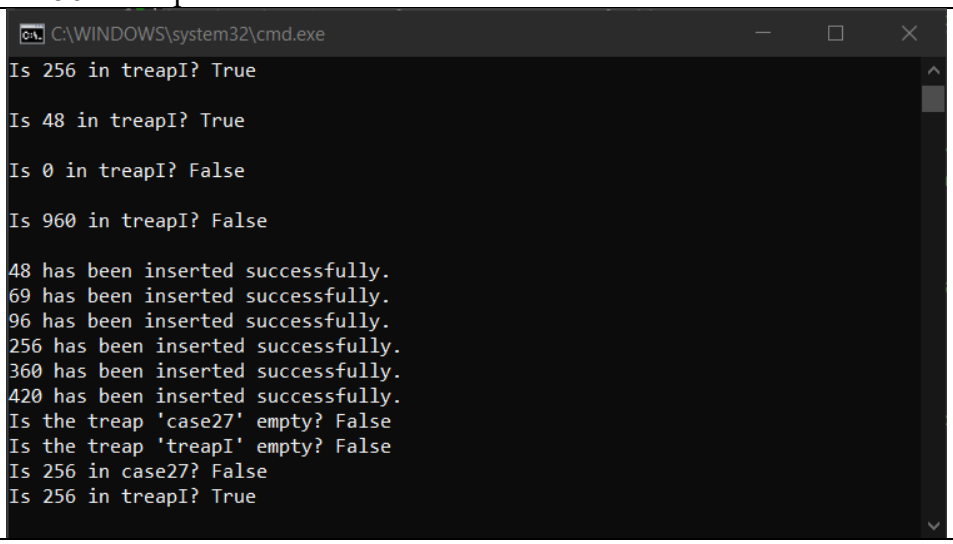
Case 22: Searching for the Root in a Treap	
Description	This case is to test if the root node can be found in a Treap without failure.
Expected Output	Is 256 in treapI? True
Actual Output	

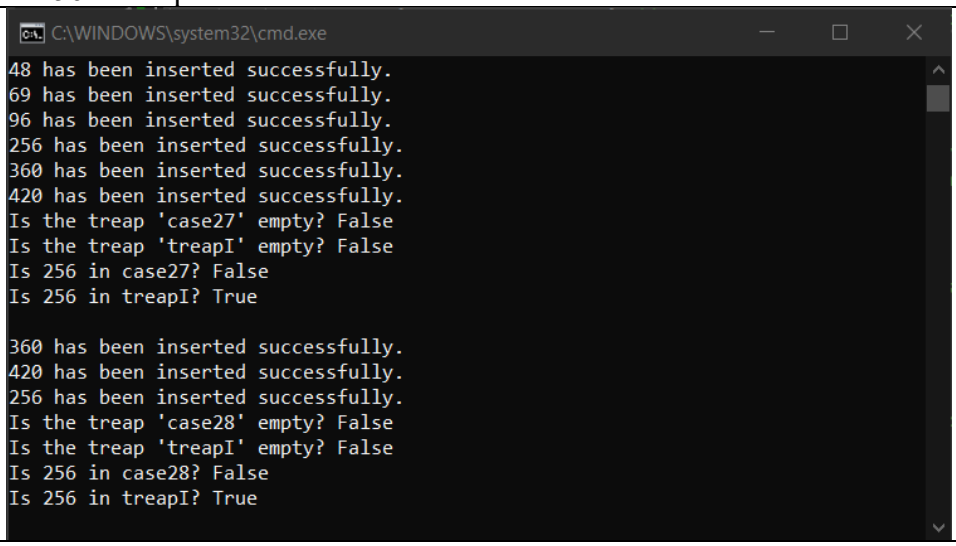
Case 23: Searching for a Leaf in a Treap	
Description	This case is to test if a leaf node can be found in a Treap without failure.
Expected Output	Is 48 in treapI? True
Actual Output	

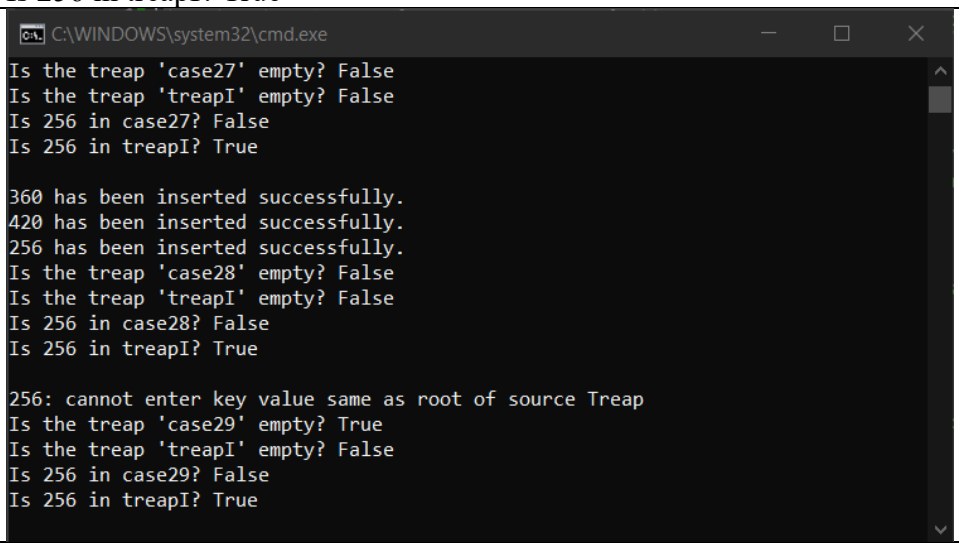
Case 24: Searching for a Node that Doesn't Exist in a Treap	
Description	This case is to test if finding a node that doesn't exist in a Treap returns null.
Expected Output	Is 0 in treapI? False
Actual Output	

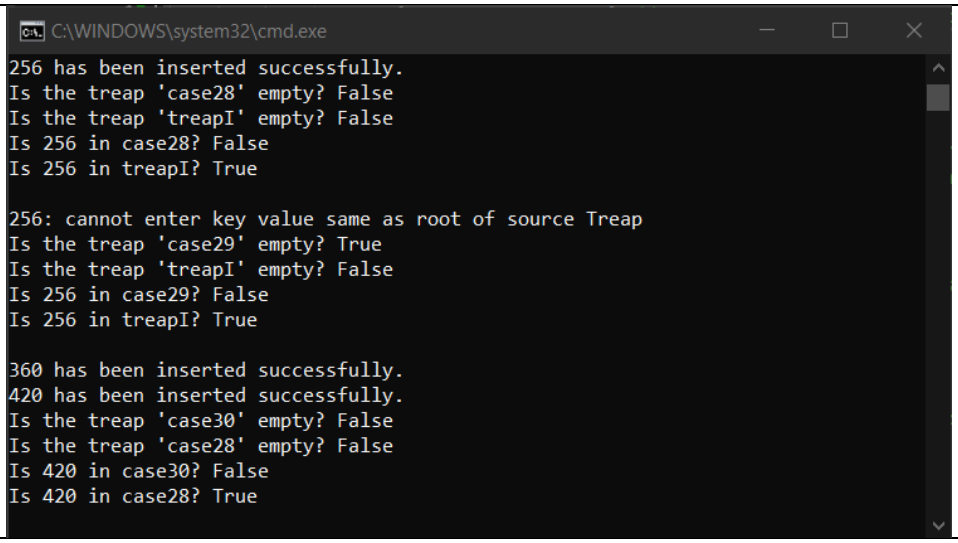
Case 25: Searching for a Node that has Already Been Deleted in a Treap	
Description	This case is to test if finding a node that has already been deleted in a Treap returns null.
Expected Output	Is 960 in treapI? False
Actual Output	

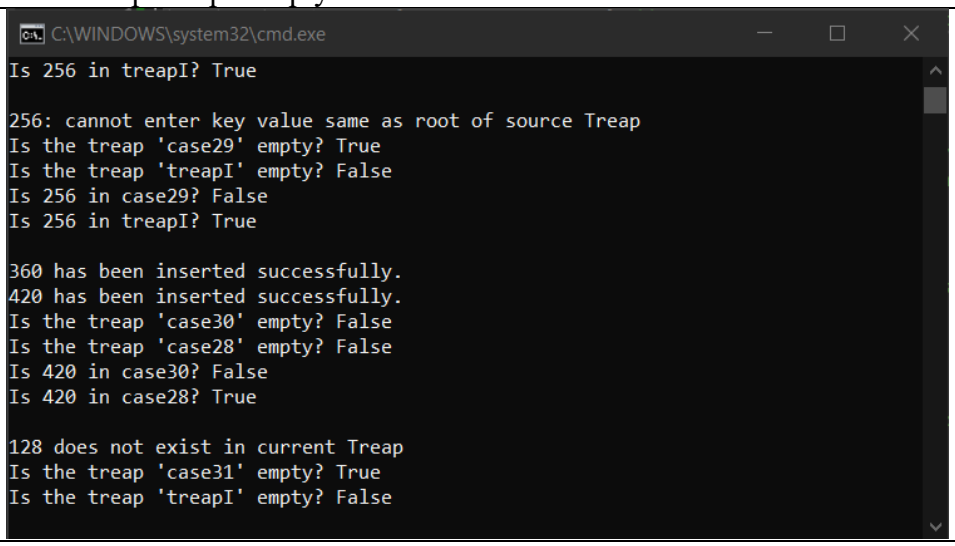
Case 26: Searching for an Invalid Node in a Treap	
Description	This case is to test if finding an invalid node in a Treap can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from 'string' to 'int'
Actual Output	

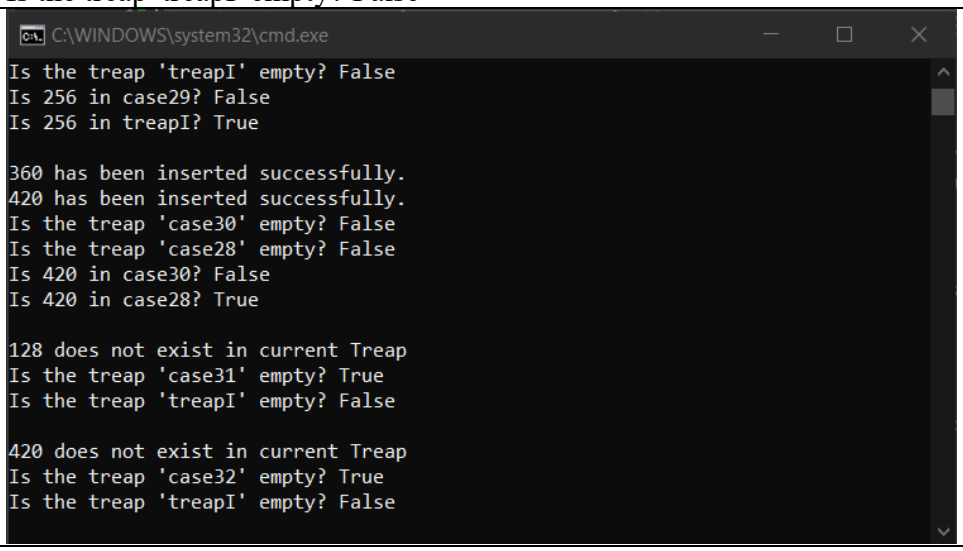
Case 27: Splitting a Treap at a Left Child	
Description	This case is to test if splitting a Treap at a left child produces two valid treaps.
Expected Output	// Insertions based on InsertP Is the treap 'case27' empty? False Is the treap 'treapI' empty? False Is 256 in case27? False Is 256 in treapI? True
Actual Output	

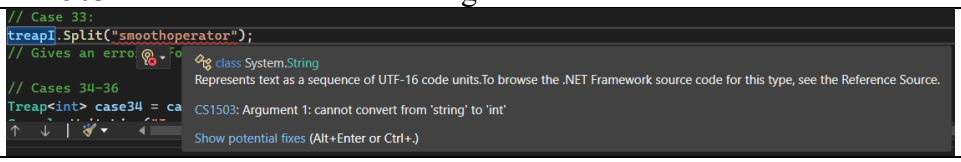
Case 28: Splitting a Treap at a Right Child	
Description	This case is to test if splitting a Treap at a right child produces two valid treaps.
Expected Output	// Insertions based on InsertP Is the treap 'case28' empty? False Is the treap 'treapI' empty? False Is 256 in case28? False Is 256 in treapI? True
Actual Output	

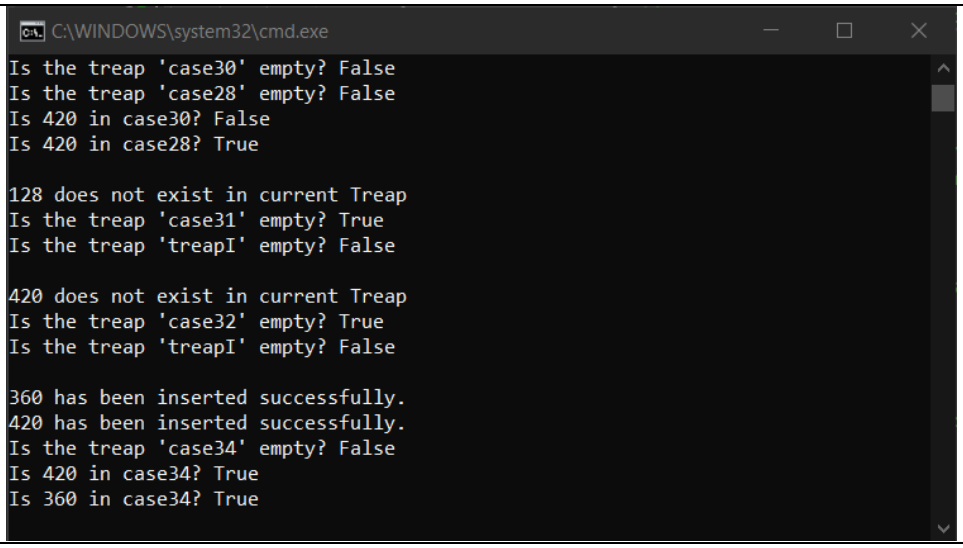
Case 29: Splitting a Treap at the Root	
Description	This case is to test if splitting a Treap at the root fails.
Expected Output	256: cannot enter key value same as root of source Treap Is the treap 'case29' empty? True Is the treap 'treapI' empty? False Is 256 in case29? False Is 256 in treapI? True
Actual Output	

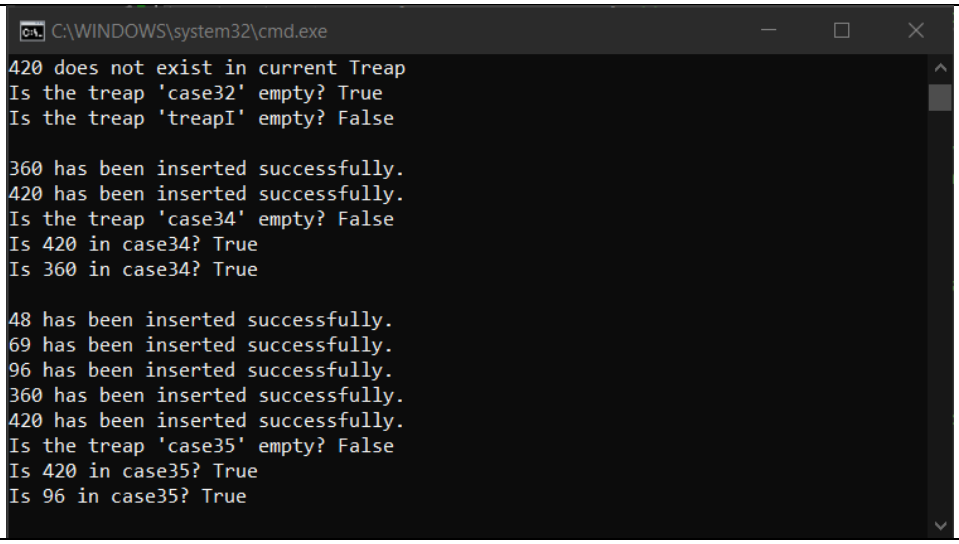
Case 30: Splitting a Treap at a Leaf	
Description	This case is to test if splitting a Treap at a leaf node produces two valid treaps.
Expected Output	// Insertions based on InsertP Is the treap 'case30' empty? False Is the treap ' case28' empty? False Is 420 in case30? False Is 420 in case28? True
Actual Output	

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	128 does not exist in current Treap Is the treap 'case31' empty? True Is the treap 'treapI' empty? False
Actual Output	

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	

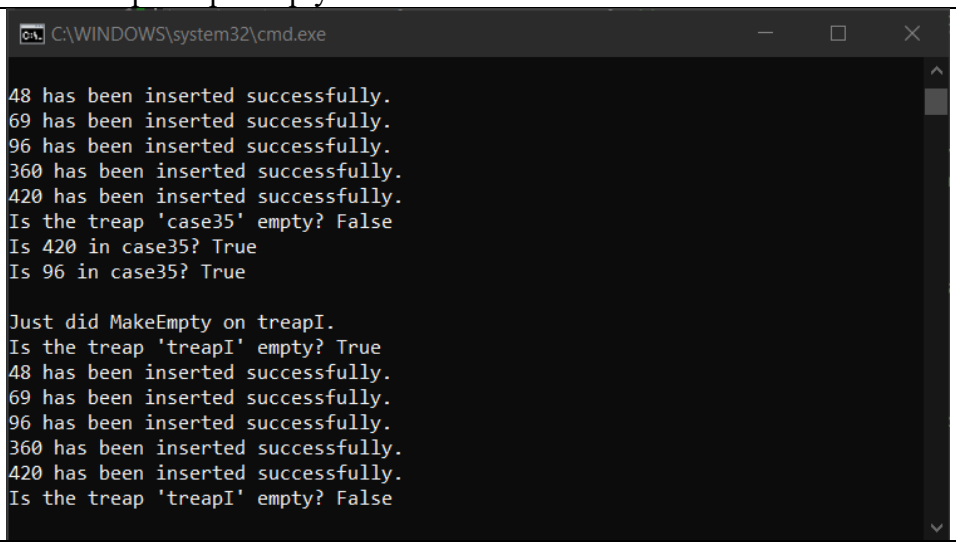
Case 33: Splitting a Treap at an Invalid Node	
Description	This case is to test if splitting an invalid node in a Treap can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from ‘string’ to ‘int’
Actual Output	

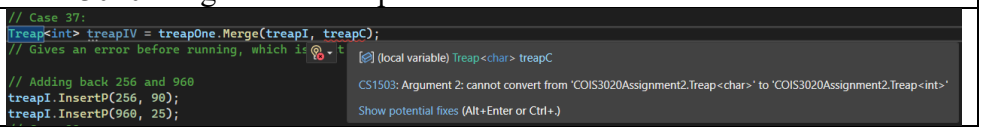
Case 34: Merging two Treaps with the Same Types and Balance	
Description	This case is to test if the merging of two treaps with the same type and balance make one valid treap.
Expected Output	// Insertions based on InsertP Is the treap 'case34' empty? False Is 420 in case34? True Is 360 in case34? True
Actual Output	

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	

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.

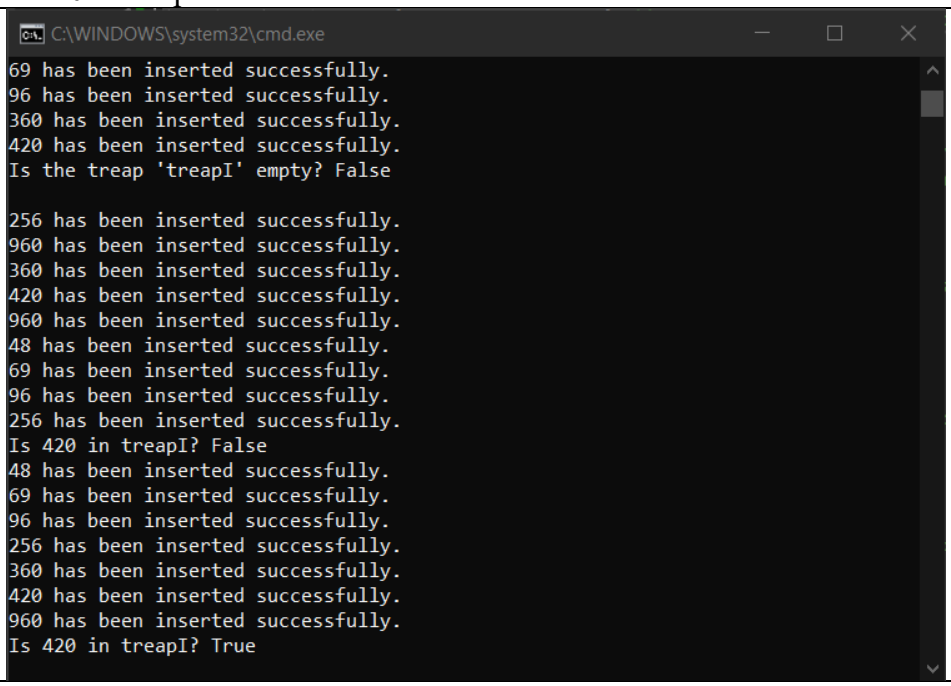
Expected Output	Just did MakeEmpty on treapI.
	Is the treap 'treapI' empty? True
	// After merge
	// Insertions based on InsertP
	Is the treap 'treapI' empty? False

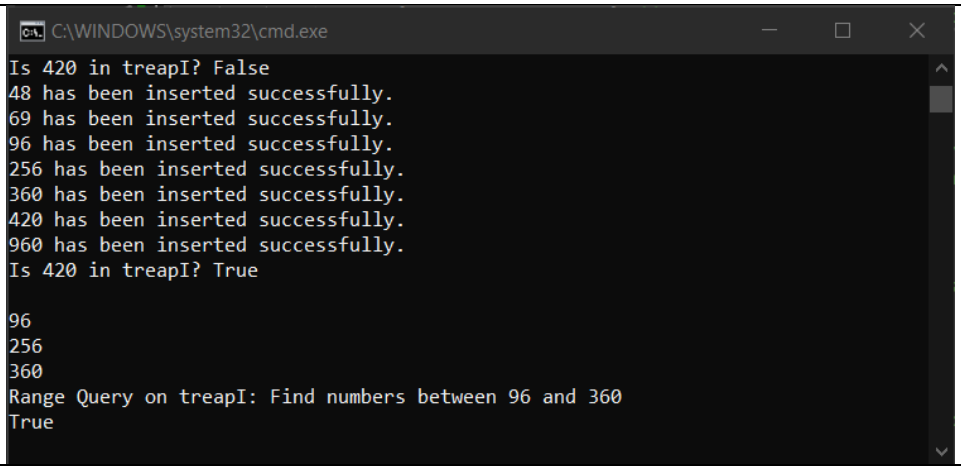
Actual Output	

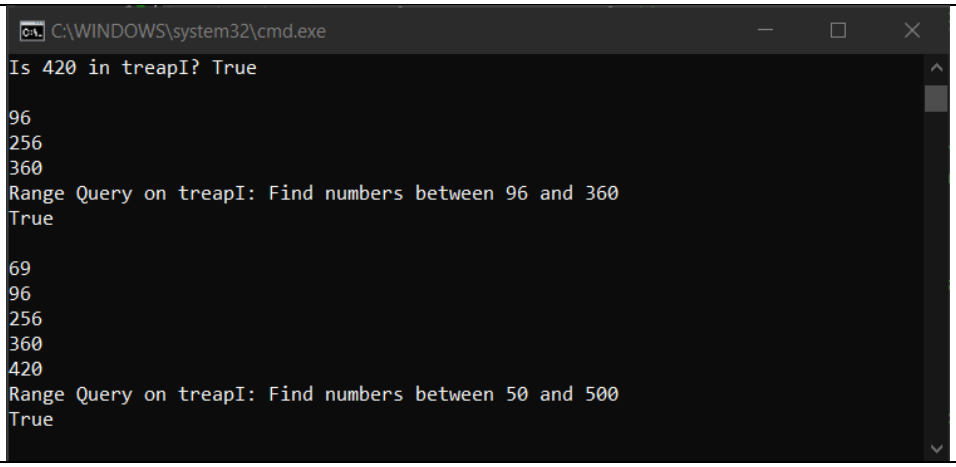
Case 37: Merging two Treaps with Different Types	
Description	This case is to test if the merging of two treaps with different types can not be done without throwing errors during compiling.
Expected Output	CS1503: cannot convert from COIS3020Assignment2.Treap<char> to COIS3020Assignment2.Treap<int>
Actual Output	

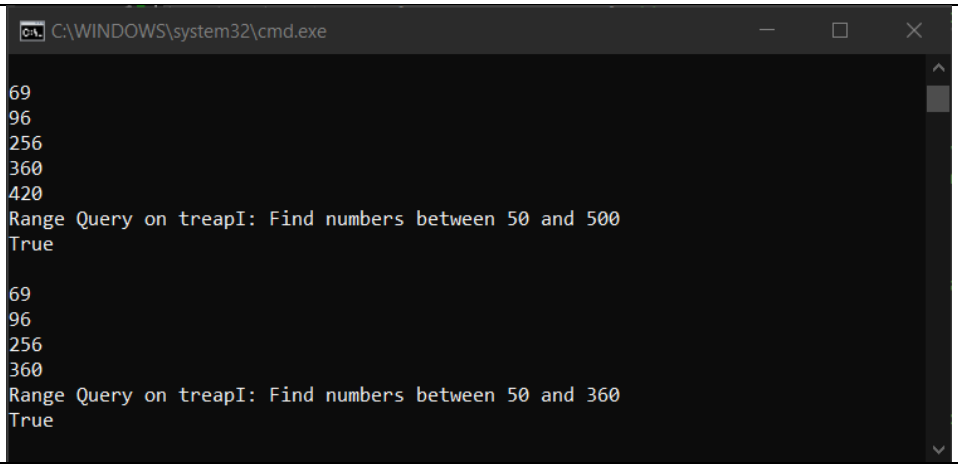
Case 38: Splitting a Treap and then Merging it back Together	
Description	This case is to test if the splitting and merging of a treap returns the same treap that it started off at.

Expected Output	// After some more insertions
	// Split
	// Insertions based on InsertP
	Is 420 in treapI? False
	// Merge
	// Insertions based on InsertP
	Is 420 in treapI? True

Actual Output	

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	

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	

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