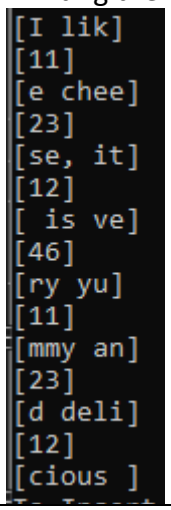
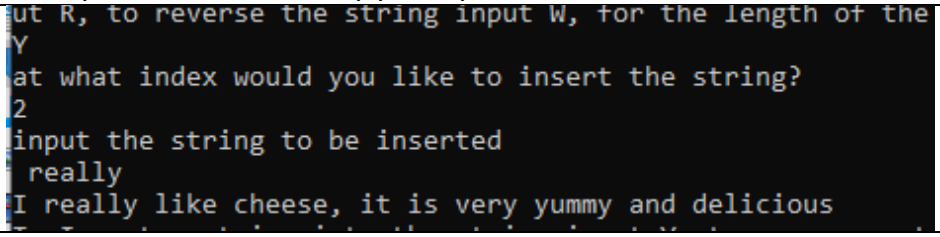
Testing Documentation

Part A: Testing with a string thas more than 10 characters(Turns into a Tree)

Test 1	
Description	Turn a string into a rope
Input	I like cheese, it is very yummy and delicious
Expected Output	A rope
Actual Output	Printing the tree in order reveals it contains 15 nodes 

Test 2	
Description	Insert a substring into the rope
Input	2, { really}
Expected Output	I really like cheese, it is very yummy and delicious
Actual Output	

Test 3	
Description	Deleting a substring from the rope
Input	3 , 9
Expected Output	I like cheese, it is very yummy and delicious

Actual Output	X From what index would you like to begin the removal? 3 From what index would you like to end the removal? 9 I like cheese, it is very yummy and delicious

Test 4	
Description	Reversing the rope
Input	I like cheese, it is very yummy and delicious
Expected Output	suoiciled dna ymmuy yrev si ti ,eseehc ekil I
Actual Output	suoiciled dna ymmuy yrev si ti ,eseehc ekil I

Test 5	
Description	Finding the substring between 2 indexes (including the indexes themselves)
Input	I like cheese, it is very yummy and delicious , 3 , 6
Expected Output	Like
Actual Output	F From what index would you like the substring to begin? 3 At what index would you like to end the substring? 6 like

Test 6	
Description	Print the length
Input	-
Expected Output	46
Actual Output	46

Test 7	
Description	Search for a character that exists (c)
Input	c
Expected Output	8
Actual Output	what character are you looking for? c 8

Test 8	
Description	Search for a character that doesn't exist
Input	Z

Expected Output	That character doesn't exist in the string
Actual Output	z That character doesnt exist in the string

Test 9	
Description	Retrieve a character at an index that doesn't exist
Input	100
Expected Output	Index doesn't exist
Actual Output	what index? 100 That index does not exist

Test 10	
Description	Retrieve a character at an index that does exist
Input	10
Expected Output	e
Actual Output	what index? 10 e

Test 11	
Description	Printing the tree
Input	-
Expected Output	A line by line Inorder print of the tree
Actual Output	[I lik] [11] [e chee] [23] [se, it] [12] [is ve] [46] [ry yu] [11] [mmy an] [23] [d deli] [12] [cious]

Add as many testing tables as you need.

Part B: Testing with a string less than 10 characters (results in a tree with 1 node)

Test 1	
Description	Turn a string into a rope
Input	cheese
Expected Output	A rope
Actual Output	Printing the tree in order reveals it contains 1 node `[cheese]`

Test 2	
Description	Insert a substring into the rope
Input	1, {good }
Expected Output	good cheese

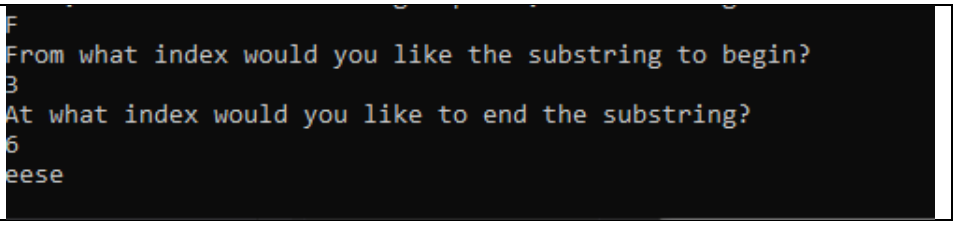
Actual Output	`at what index would you like to insert the string?`
	`1`
	`input the string to be inserted`
	`good`
	`good cheese`

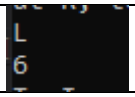
Test 3	
Description	Deleting a substring from the rope
Input	1, 5
Expected Output	cheese

Actual Output	`X`
	`From what index would you like to begin the removal?`
	`1`
	`From what index would you like to end the removal?`
	`5`
	`cheese`

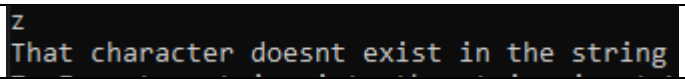
Test 4	
Description	Reversing the rope
Input	-
Expected Output	eseehc
Actual Output	`eseehc`

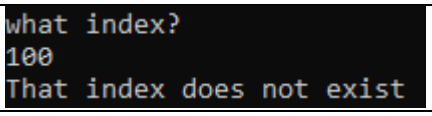
Test 5	
Description	Finding the substring between 2 indexes (including the indexes themselves)
Input	cheese, 3 , 6
Expected Output	eese

Actual Output	

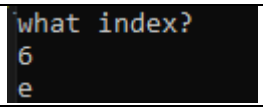
Test 6	
Description	Print the length
Input	-
Expected Output	6
Actual Output	

Test 7	
Description	Search for a character that exists (c)
Input	s
Expected Output	5
Actual Output	

Test 8	
Description	Search for a character that doesn't exist
Input	Z
Expected Output	That character doesn't exist in the string
Actual Output	

Test 9	
Description	Retrieve a character at an index that doesn't exist
Input	100
Expected Output	Index doesn't exist
Actual Output	

Test 10	
Description	Retrieve a character at an index that does exist
Input	6

Expected Output	e
Actual Output	

Test 11	
Description	Printing the tree
Input	-
Expected Output	It prints a single node
Actual Output