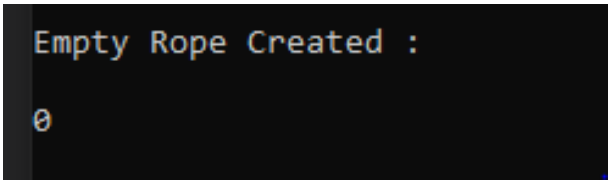
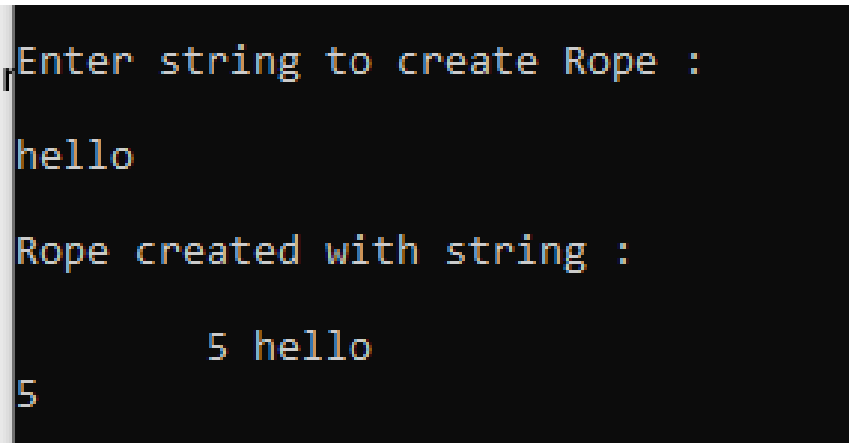
Testing Documentation

All the functions work (finally), and I tried to not use char at functions for split and tried to use originals. Almost all methods have their private method counterpart and hence there is like two of each methods. Other than that, the print method can be seen literally after each test case when I print the rope.

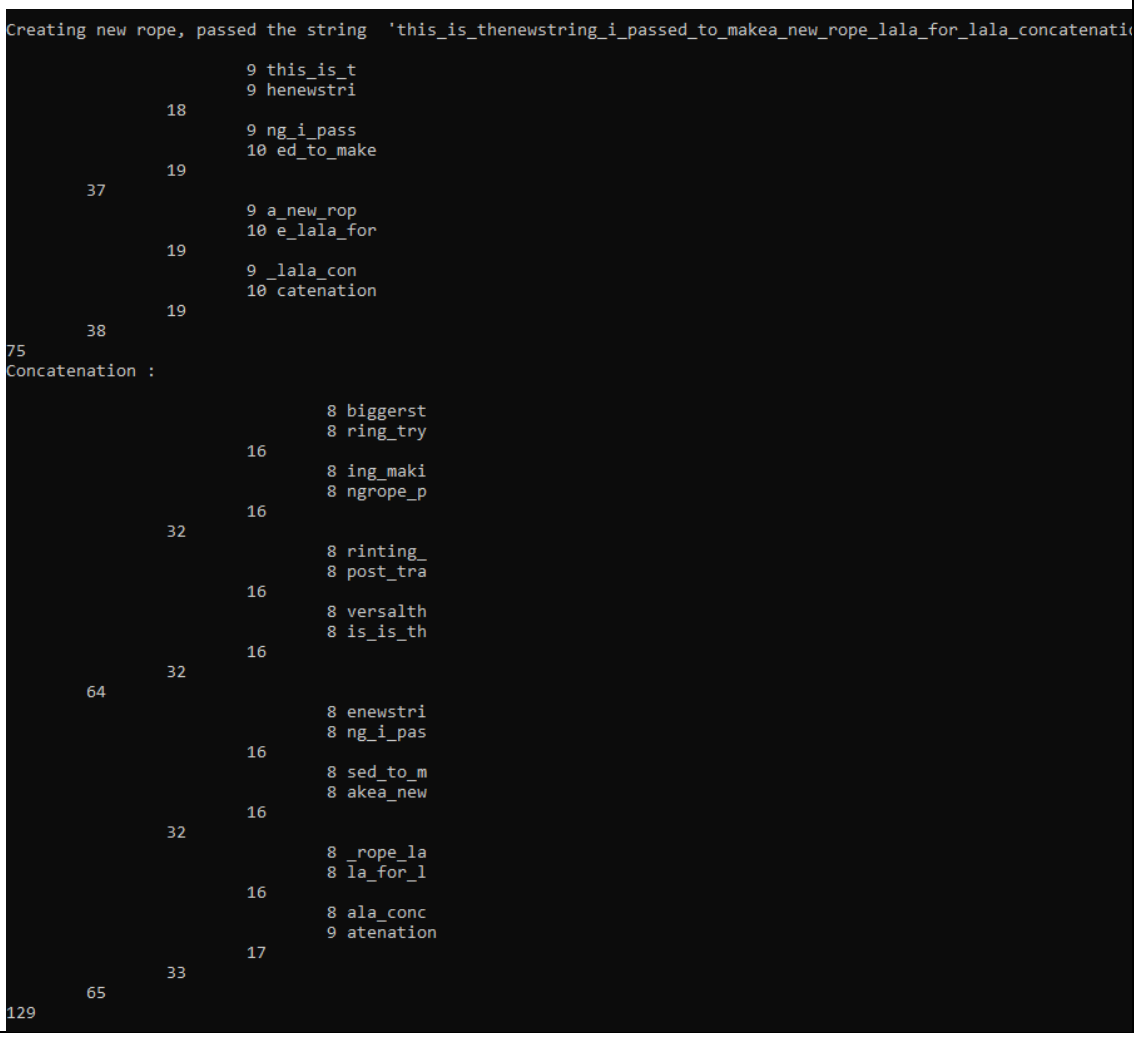
The optimizations have been done and can be seen in these test cases. In split, the final rope is compressed when print. For total string length of siblings less than 5, the nodes are merged. There is a rebalance method that is used in a few methods (like split, insert) to rebalance the rope.

Test 1	
Input	Creating an empty Rope – **Rope() constructor**
Output	

Test 2	
Input	Creating a rope for a given string – **Rope(String S) constructor**
Output	String length is less than =5 : print rope is representing rope in **HORIZONTAL** direction This includes one of the optimizations : 

Bigger string :

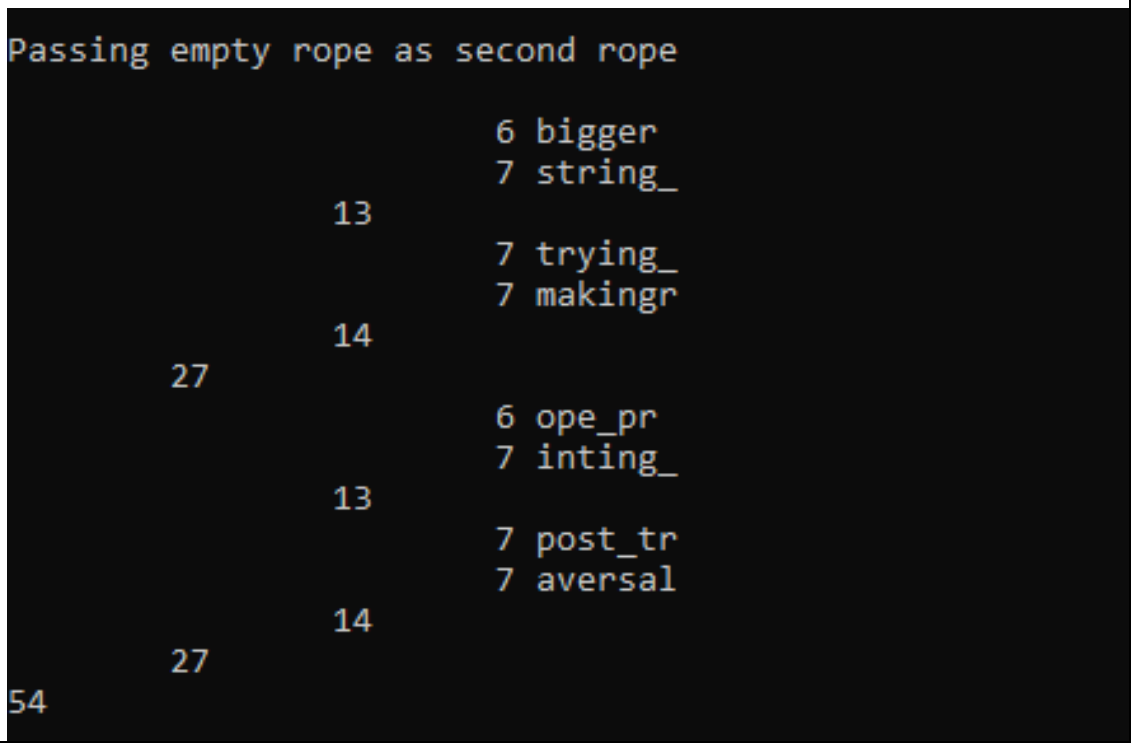
```
Empty Rope Created :

0

Enter string to create Rope :

biggerstring_trying_makingrope_printing_post_traversal

Rope created with string :

                        6 bigger
                        7 string_
                13
                        7 trying_
                        7 makingr
                14
        27
                        6 ope_pr
                        7 inting_
                13
                        7 post_tr
                        7 aversal
                14
        27
54

########## Assignment 2 Rope ############

Choose a number for various method actions (Print Method is being test
```

Test 3	
Input	Concatenating 2 ropes R1 and R2 where R1 and R2 are not null : **Check PRINT method as well**
Output	Creating a rope and then concatenating the existing rope with the previous rope 

```
Creating new rope, passed the string  'this_is_thenewstring_i_passed_to_makea_new_rope_lala_for_lala_concatenati

                        9 this_is_t
                        9 henewstri
                18
                        9 ng_i_pass
                        10 ed_to_make
                19
        37
                        9 a_new_rop
                        10 e_lala_for
                19
                        9 _lala_con
                        10 catenation
                19
        38
75
Concatenation :

                                8 biggerst
                                8 ring_try
                        16
                                8 ing_maki
                                8 ngrope_p
                        16
                32
                                8 rinting_
                                8 post_tra
                        16
                                8 versalth
                                8 is_is_th
                        16
                32
        64
                                8 enewstri
                                8 ng_i_pas
                        16
                                8 sed_to_m
                                8 akea_new
                        16
                32
                                8 _rope_la
                                8 la_for_l
                        16
                                8 ala_conc
                                9 atenation
                        17
                33
        65
129
```

Test 4	
Input	Concatenating 2 ropes R1 and R2 where R1 is null
Output	

```
########## Assignment 2 Rope ###########

Choose a number for various method actions (Print Method is being tested in literally every step :

 1. Concatenate 2 ropes with 3 conditions - first rope null, second rope null, no ropes null
 2. Split rope at index i
 3. Insert string S entered by user at index i, at end and at begining
 4. Delete a substring between 2 indices
 5. Substring method to get substring in between 2 indices
 6. Return character at index i charat
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
1

 ############# Concatenate Method ###########


Three options :


We will use the empty rope we created in the begining


Passing empty rope as first rope

                        6 bigger
                        7 string_
                13
                        7 trying_
                        7 makingr
                14
        27
                        6 ope_pr
                        7 inting_
                13
                        7 post_tr
                        7 aversal
                14
        27
54
```

Test 5	
Input	Concatenating 2 ropes R1 and R2 where R2 is null
Output	

Test 6	
Input	Splitting a rope at index i
Output	At index : 8 : Optimization is taken care of

```
biggerstring_trying_makingrope_printing_post_traversal

Rope created with string :

                        6 bigger
                        7 string_
                13
                        7 trying_
                        7 makingr
                14
        27
                        6 ope_pr
                        7 inting_
                13
                        7 post_tr
                        7 aversal
                14
        27
54

########## Assignment 2 Rope ###########

Choose a number for various method actions (Print Method is being tested in literally every st

 1. Concatenate 2 ropes with 3 conditions - first rope null, second rope null, no ropes null
 2. Split rope at index i
 3. Insert string S entered by user at index i, at end and at begining
 4. Delete a substring between 2 indices
 5. Substring method to get substring in between 2 indices
 6. Return character at index i charat
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
2

 ############# Split Method ###########


 Enter the index you want to split at

8
        4 bigg
        5 erstr
9
                        5 ing_t
                        6 rying_
                11
                        5 makin
                        6 grope_
                11
        22
                        5 print
                        6 ing_po
                11
                        6 st_tra
                        6 versal
                12
        23
45
```

Test 7	
Input	Splitting a rope at index i
Output	At index : 0 : Optimization is taken care of

```
Enter string to create Rope :

biggerstring_trying_makingrope_printing_post_traversal

Rope created with string :

                        6 bigger
                        7 string_
                13
                        7 trying_
                        7 makingr
                14
        27
                        6 ope_pr
                        7 inting_
                13
                        7 post_tr
                        7 aversal
                14
        27
54

########## Assignment 2 Rope ###########

Choose a number for various method actions (Print Method is being tested in literally every step

 1. Concatenate 2 ropes with 3 conditions - first rope null, second rope null, no ropes null
 2. Split rope at index i
 3. Insert string S entered by user at index i, at end and at begining
 4. Delete a substring between 2 indices
 5. Substring method to get substring in between 2 indices
 6. Return character at index i charat
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
2

 ############# Split Method ###########


 Enter the index you want to split at

0
        1 b
1
                        6 iggers
                        7 tring_t
                13
                        6 rying_
                        7 makingr
                13
        26
                        6 ope_pr
                        7 inting_
                13
                        7 post_tr
                        7 aversal
                14
        27
53
```

Test 8	
Input	Inserting a string S in between index i
Output	At index = 23; string = read

```
 10. Return string represented by current rope
3

 ############# Insert Method ###########


 Enter the index you want to enter at

23

Enter the string to be inserted

read
                        7 biggers
                        7 tring_t
                14
                        7 rying_m
                        8 akireadn
                15
        29
                        7 grope_p
                        7 rinting
                14
                        7 _post_t
                        8 raversal
                15
        29
58
```

Test 9	
Input	Inserting a string S at end of the string
Output	At index = 67 (input) now 67 is greater than the length of existing string so at end it will be inserted. String = read

```
 9. Length of string
 10. Return string represented by current rope
3

 ############## Insert Method ############


 Enter the index you want to enter at

67

Enter the string to be inserted

read
                        7 biggers
                        7 tring_t
                14
                        7 rying_m
                        8 akingrop
                15
        29
                        7 e_print
                        7 ing_pos
                14
                        7 t_trave
                        8 rsalread
                15
        29
58
```

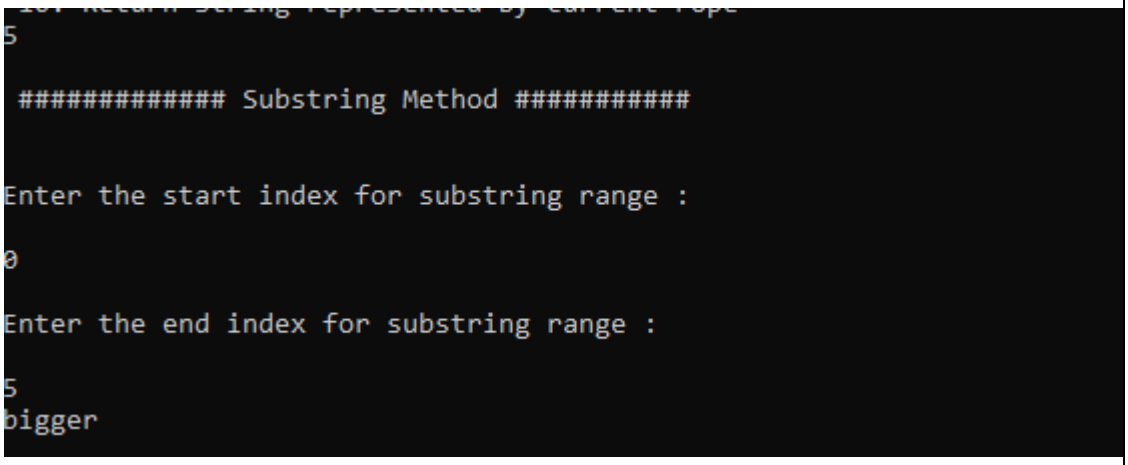
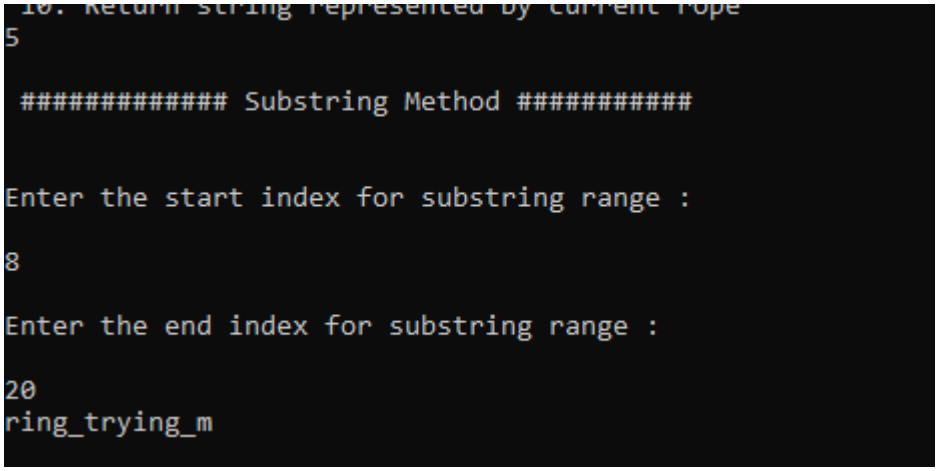
Test 10	
Input	Inserting a String at the beginning
Output	At index = 0 ; string = read

```
Enter the string to be inserted

read
                        7 readbig
                        7 gerstri
                14
                        7 ng_tryi
                        8 ng_makin
                15
        29
                        7 grope_p
                        7 rinting
                14
                        7 _post_t
                        8 raversal
                15
        29
58
```

Test 11	
Input	Deleting a substring
Output	From index = 20, end index = 5 ➔ 20+5 = 25

```
########## Assignment 2 Rope ###########

Choose a number for various method actions (Print Method is being tested in literally every step :

 1. Concatenate 2 ropes with 3 conditions - first rope null, second rope null, no ropes null
 2. Split rope at index i
 3. Insert string S entered by user at index i, at end and at begining
 4. Delete a substring between 2 indices
 5. Substring method to get substring in between 2 indices
 6. Return character at index i charat
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
4

 ############# Delete Method ###########


Enter the start index for substring range to be deleted :

20

Enter the end index for substring range to be deleted :

5
                        6 bigger
                        6 string
                12
                        6 _tryin
                        6 g_mrop
                12
        24
                        6 e_prin
                        6 ting_p
                12
                        6 ost_tr
                        7 aversal
                13
        25
49
```

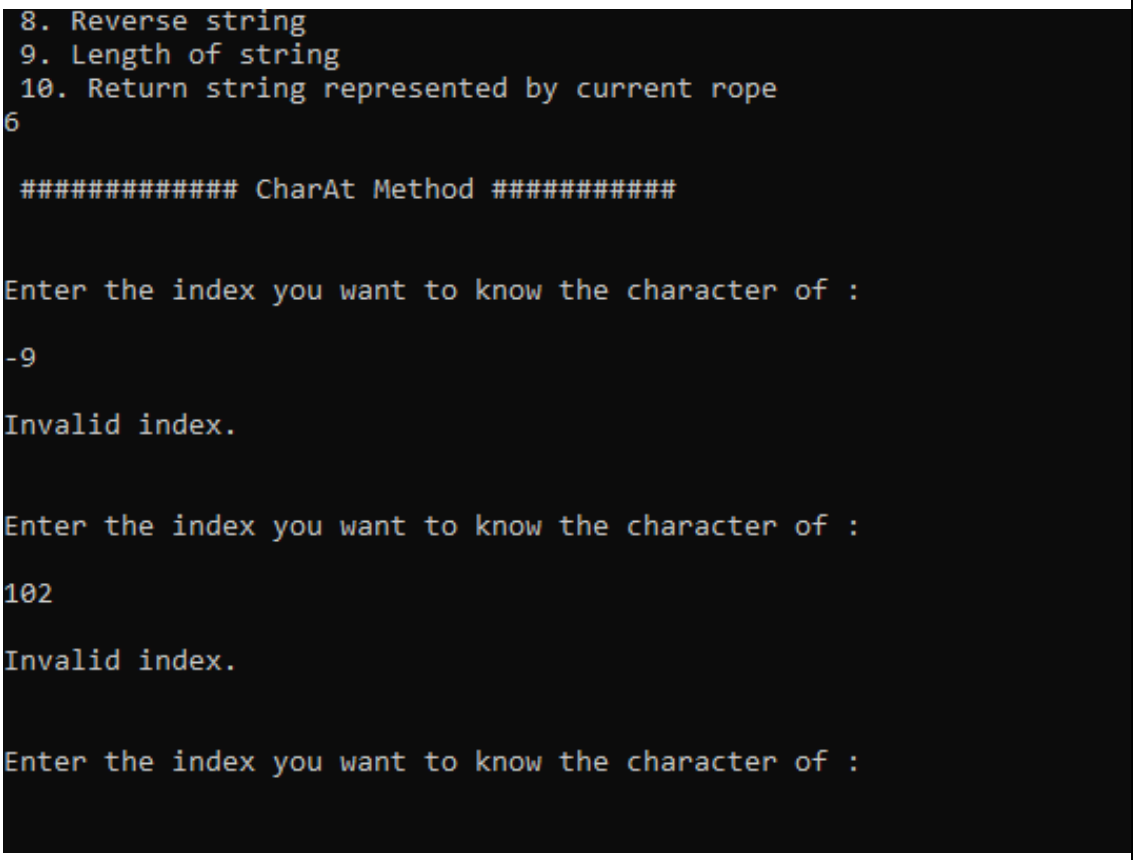
Test 12	
Input	Deleting a substring : Invalid Indices
Output	9. Length of string 10. Return string represented by current rope 4 ############ Delete Method ########### Enter the start index for substring range to be deleted : -4 Enter the end index for substring range to be deleted : 90 Invalid indices. Enter the start index for substring range to be deleted : 13 Enter the end index for substring range to be deleted : 13 Invalid indices. Enter the start index for substring range to be deleted :

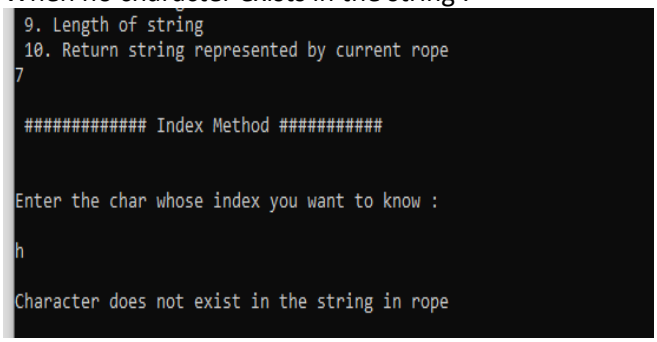
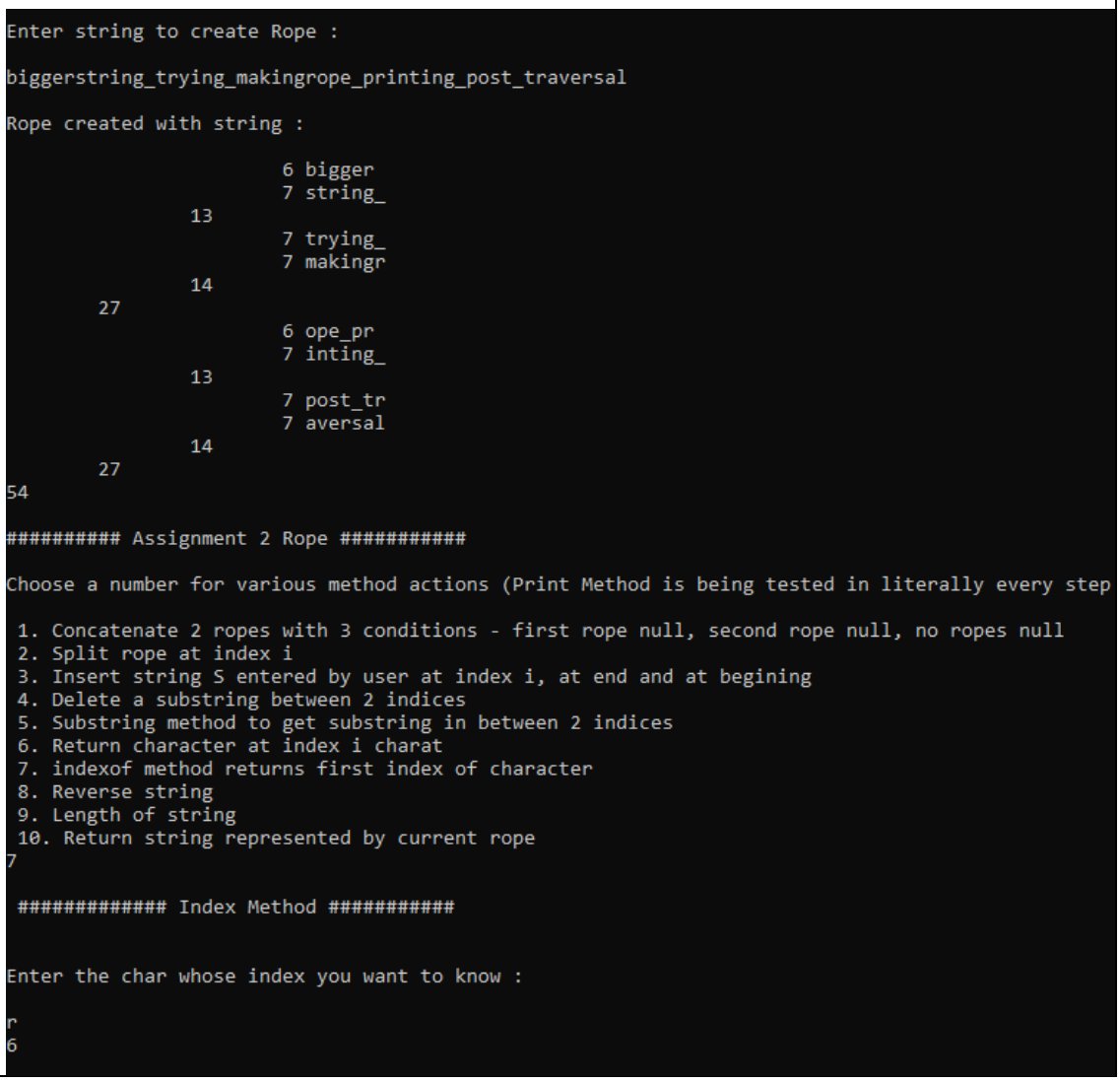
Test 13	
Input	Substring method returns substring between two indices
Output	Between 0 and 5  Between 8 and 20  Invalid Indices

```
 10. Return string represented by current rope
5

 ############# Substring Method ###########


Enter the start index for substring range :

-5

Enter the end index for substring range :

333

Invalid indices.


Enter the start index for substring range :
```

Test 14	
Input	Returning a character at index i
Output	

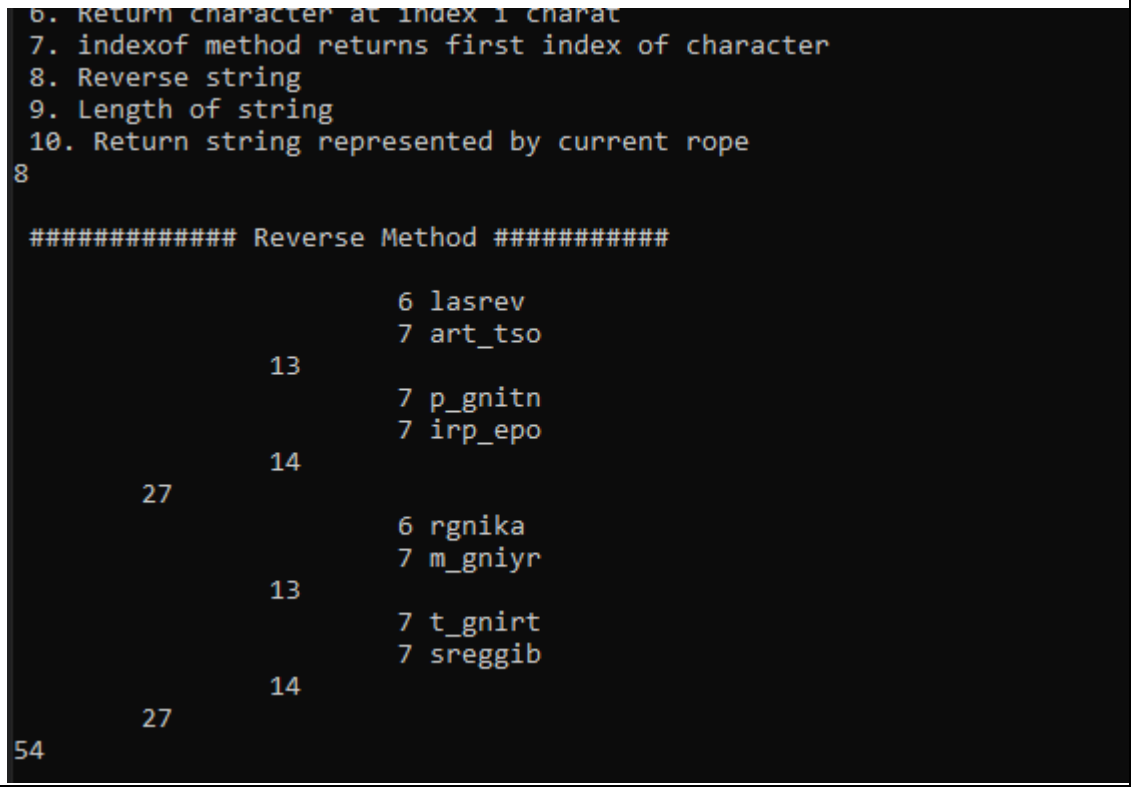
```
Rope created with string :

                        6 bigger
                        7 string_
                13
                        7 trying_
                        7 makingr
                14
        27
                        6 ope_pr
                        7 inting_
                13
                        7 post_tr
                        7 aversal
                14
        27
54

########## Assignment 2 Rope ###########

Choose a number for various method actions (Print Method is being tested in literally every ste

 1. Concatenate 2 ropes with 3 conditions - first rope null, second rope null, no ropes null
 2. Split rope at index i
 3. Insert string S entered by user at index i, at end and at begining
 4. Delete a substring between 2 indices
 5. Substring method to get substring in between 2 indices
 6. Return character at index i charat
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
6

 ############# CharAt Method ###########


Enter the index you want to know the character of :

23
i
```

Test 15	
Input	Returning a character at index I – invalid input
Output	

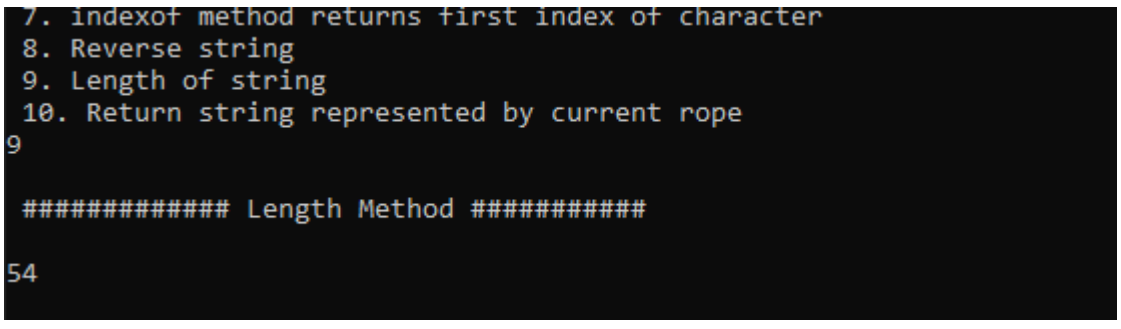
Test 16	
Input	Returning first index a character is at
Output	When no character exists in the string :  When the character exists 

```
 9. Length of string
 10. Return string represented by current rope
7

 ############ Index Method ###########


Enter the char whose index you want to know :

h

Character does not exist in the string in rope
```

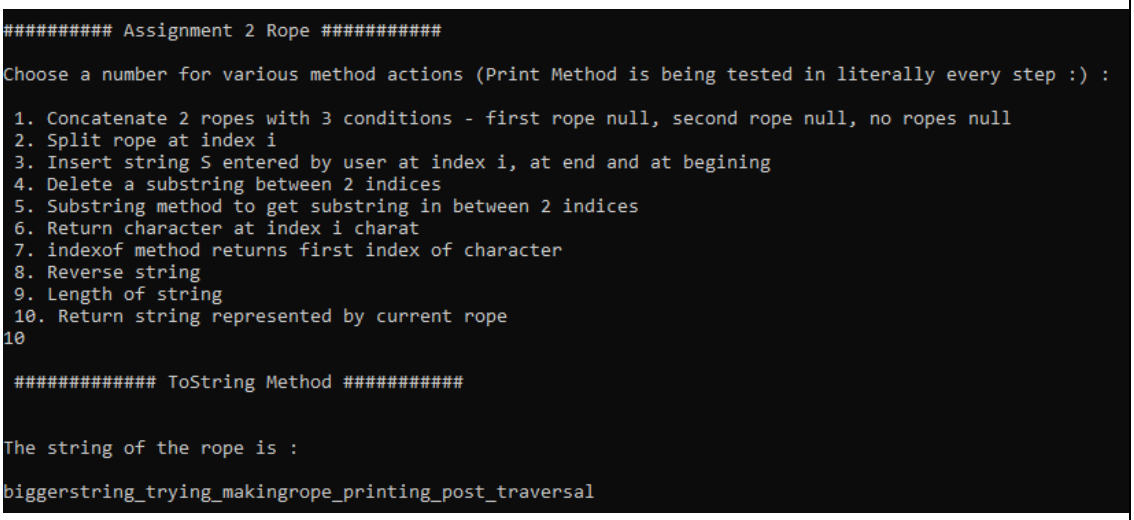
```
Enter string to create Rope :

biggerstring_trying_makingrope_printing_post_traversal

Rope created with string :

                        6 bigger
                        7 string_
                13
                        7 trying_
                        7 makingr
                14
        27
                        6 ope_pr
                        7 inting_
                13
                        7 post_tr
                        7 aversal
                14
        27
54

########## Assignment 2 Rope ###########

Choose a number for various method actions (Print Method is being tested in literally every step

 1. Concatenate 2 ropes with 3 conditions - first rope null, second rope null, no ropes null
 2. Split rope at index i
 3. Insert string S entered by user at index i, at end and at begining
 4. Delete a substring between 2 indices
 5. Substring method to get substring in between 2 indices
 6. Return character at index i charat
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
7

 ############# Index Method ###########


Enter the char whose index you want to know :

r
6
```

Test 17	
Input	Reversing the string
Output	

```
 6. Return character at index i charAt
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
8

 ############# Reverse Method ###########

                        6 lasrev
                        7 art_tso
                13
                        7 p_gnitn
                        7 irp_epo
                14
        27
                        6 rgnika
                        7 m_gniyr
                13
                        7 t_gnirt
                        7 sreggib
                14
        27
54
```

Test 18	
Input	Length of the string
Output	

```
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
9

 ############# Length Method ###########

54
```

Test 19	
Input	Returning the string represented by current rope
Output	

```
########## Assignment 2 Rope ###########

Choose a number for various method actions (Print Method is being tested in literally every step :) :

 1. Concatenate 2 ropes with 3 conditions - first rope null, second rope null, no ropes null
 2. Split rope at index i
 3. Insert string S entered by user at index i, at end and at begining
 4. Delete a substring between 2 indices
 5. Substring method to get substring in between 2 indices
 6. Return character at index i charat
 7. indexof method returns first index of character
 8. Reverse string
 9. Length of string
 10. Return string represented by current rope
10

 ############# ToString Method ###########


The string of the rope is :

biggerstring_trying_makingrope_printing_post_traversal
```