Rajahow Parajuli

* Approach, Design & Algorithm:
* I used the same binary tree from the bonus assignment some changes to insert and the tree node.
* To de code the mose code the iterator recursively calls the right or the left node depending on if it is a . or a –
* After getting each letter in word it joins it and adds it to a string
* After all the word have been converted it is returned.
* Test Plan & Test Cases:

Pictures for cases are below the table.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Actual output | expected | passed |
| - / .... / . / .-.. / .-.. / --- | T h e l l o | T h e l l o | yes |
| - .... . / .-.. .-.. --- | the llo | the llo | yes |
| - .... . .-.. .-.. --- | thello | thello | yes |

Highlights and assumptions:

The assumptions I made for this program are:

* The user is not going to use unique characters
* The user only expects to have lowercase alpha results.

The highlights:

I learned how to use binary tree in a program