Diploma Software Testing

Robert

30018755

Diploma Software Testing

Programming3\_Act2\_BinaryTree

# Table of Contents

[Table of Contents](#_Toc65787991)

[Introduction and Purpose. 1](#_Toc65787992)

[UI Elements 2](#_Toc65787993)

[Testing Tables 4](#_Toc65787994)

## Introduction and Purpose.

The purpose of this document is to outline and demonstrate the use and testing process of the application. The application being tested is a demonstration of a working self-balancing AVL tree in a Console Environment.

AVL Tree’s are a collection of nodes that contain a key and some data.

The key is used to determine where in the tree the node should be located and also to interact with the node like an index.

If the tree is too heavy on one side (the height of the tree favors one side over the other) the Tree will rotate in a direction to balance the tree and select a new Root node.

The Root Node is the entry point for new nodes as well as the main point of interaction between the user and the application.

This application contains both a tutorial to show off its features as well as a looping application controlled by the user to insert, delete, find, update and fill X amount of new nodes as an example.

## UI Elements

The Console Application does not have a UI Element but screen shots of the terminal displaying the application can be provided.

Figure 1. User chooses ‘y’ to being tutorial which allows the program to fill up 15 empty nodes into the tree (currently unbalanced).

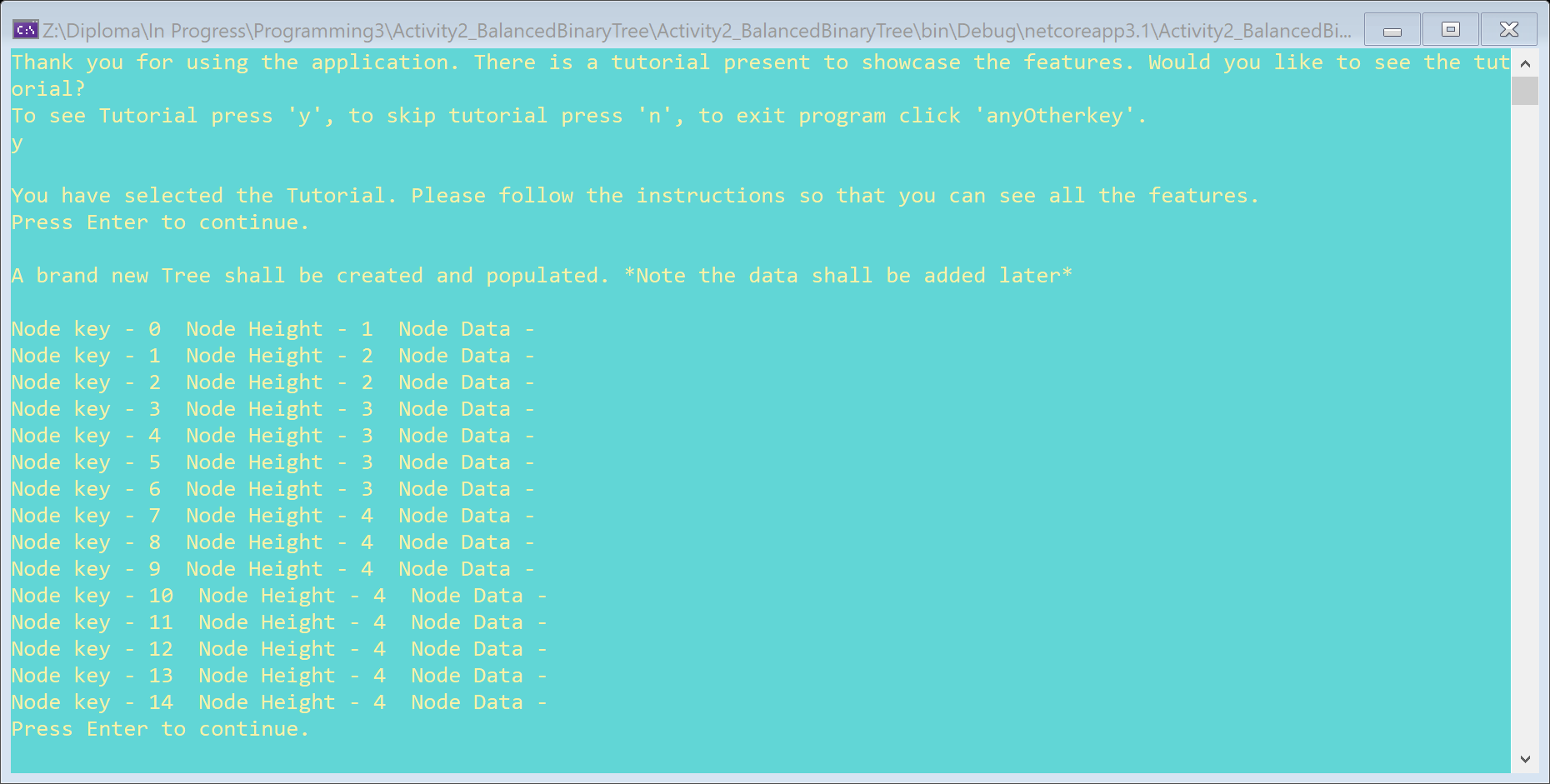


Figure 2. The Tree is then balanced. The root node the first entry (Key 7) at the top of the tree (height 4). The Lowest nodes on the tree (Keys - 0, 1, 4, 6, 8, 10, 12, 14) are at the bottom (height 1).

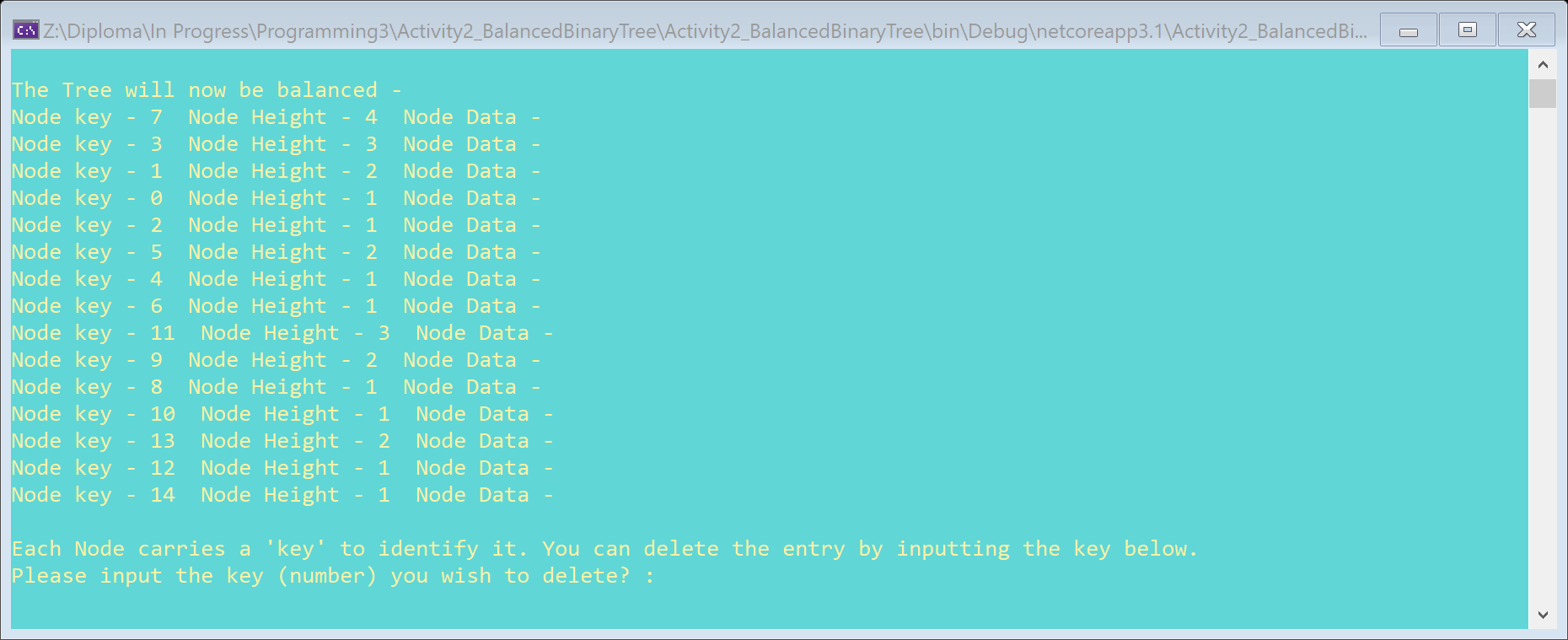


Figure 3. Allows the User to delete an entry (via the key) from the tree. The Tree is then rebalanced after the node is removed. The Tree is Rebalanced after each method.

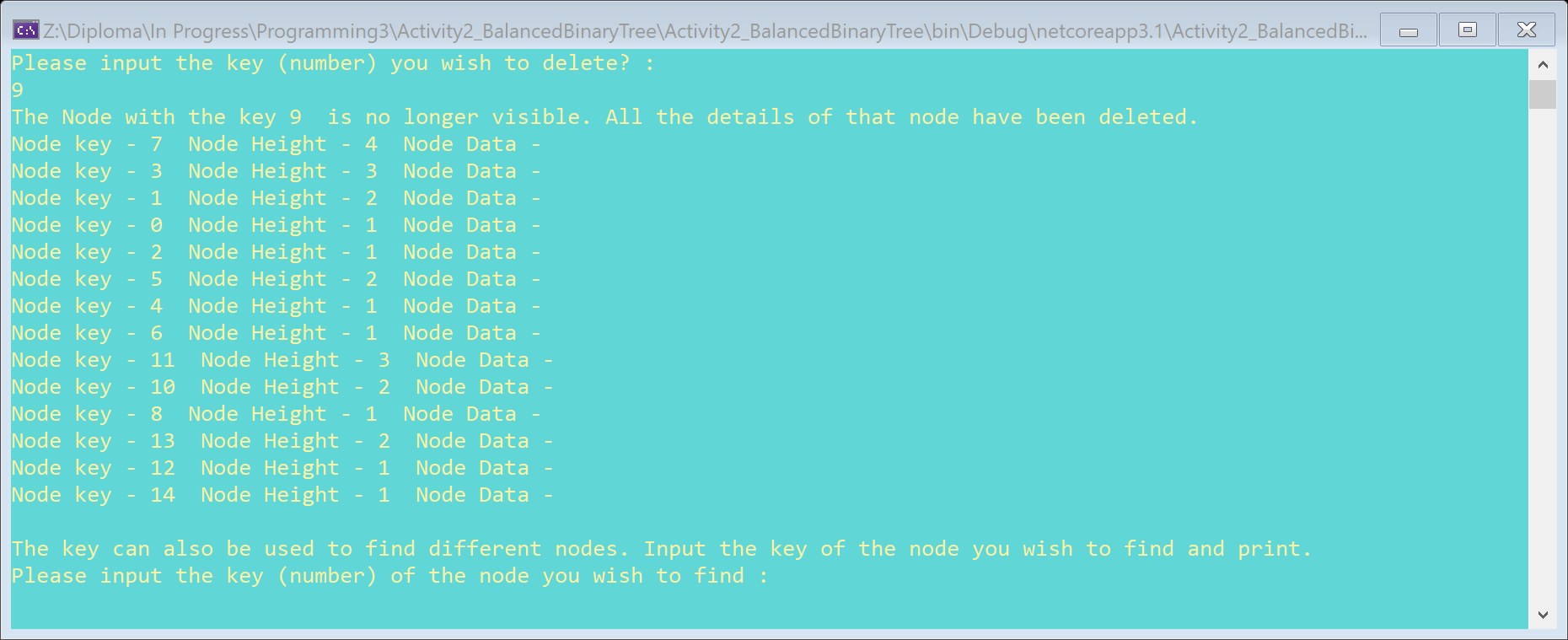


Figure 4. Allows the user to find an entry (via the key) from the tree. This function is used to manipulate a single node (delete or update). The Tree is Rebalanced after each method.

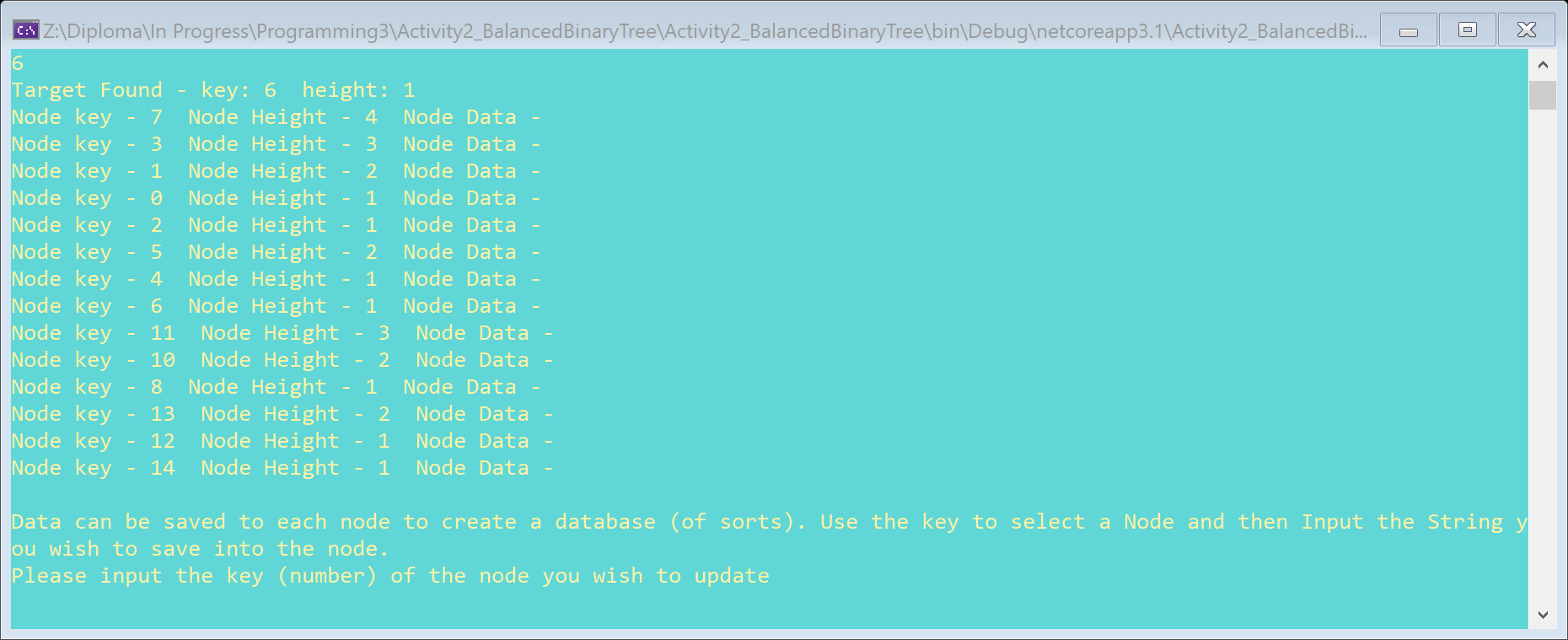


Figure 5. Allows the user to find a node and update it with a String value that is also input by the user. Node 14 is chosen and the String “This IS an Example” is input into the node. The Tree is Rebalanced after each method.

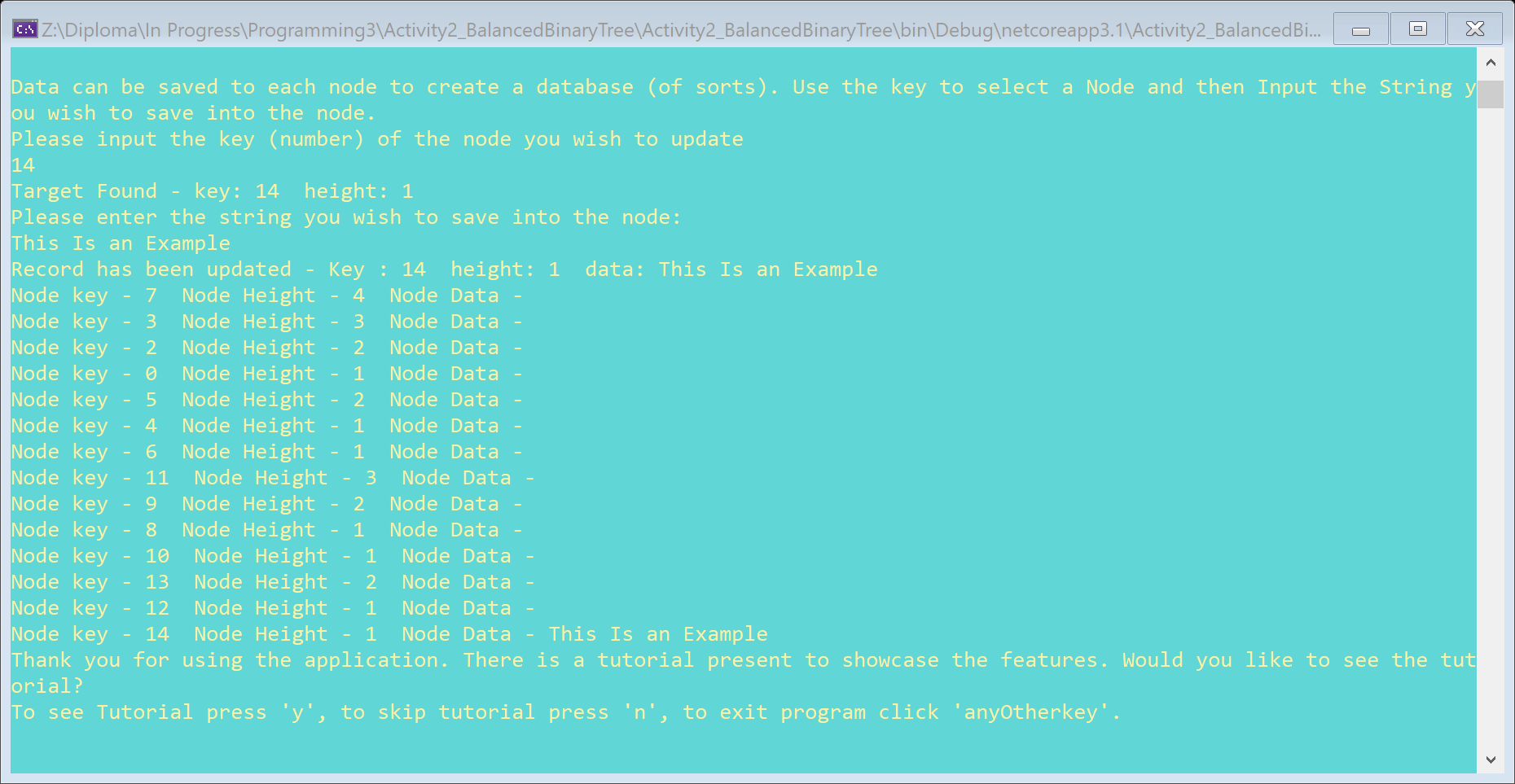


Figure 6. After the Tutorial has ended the user is able to continue to use the application as a loop for faster and user friendly functionality.

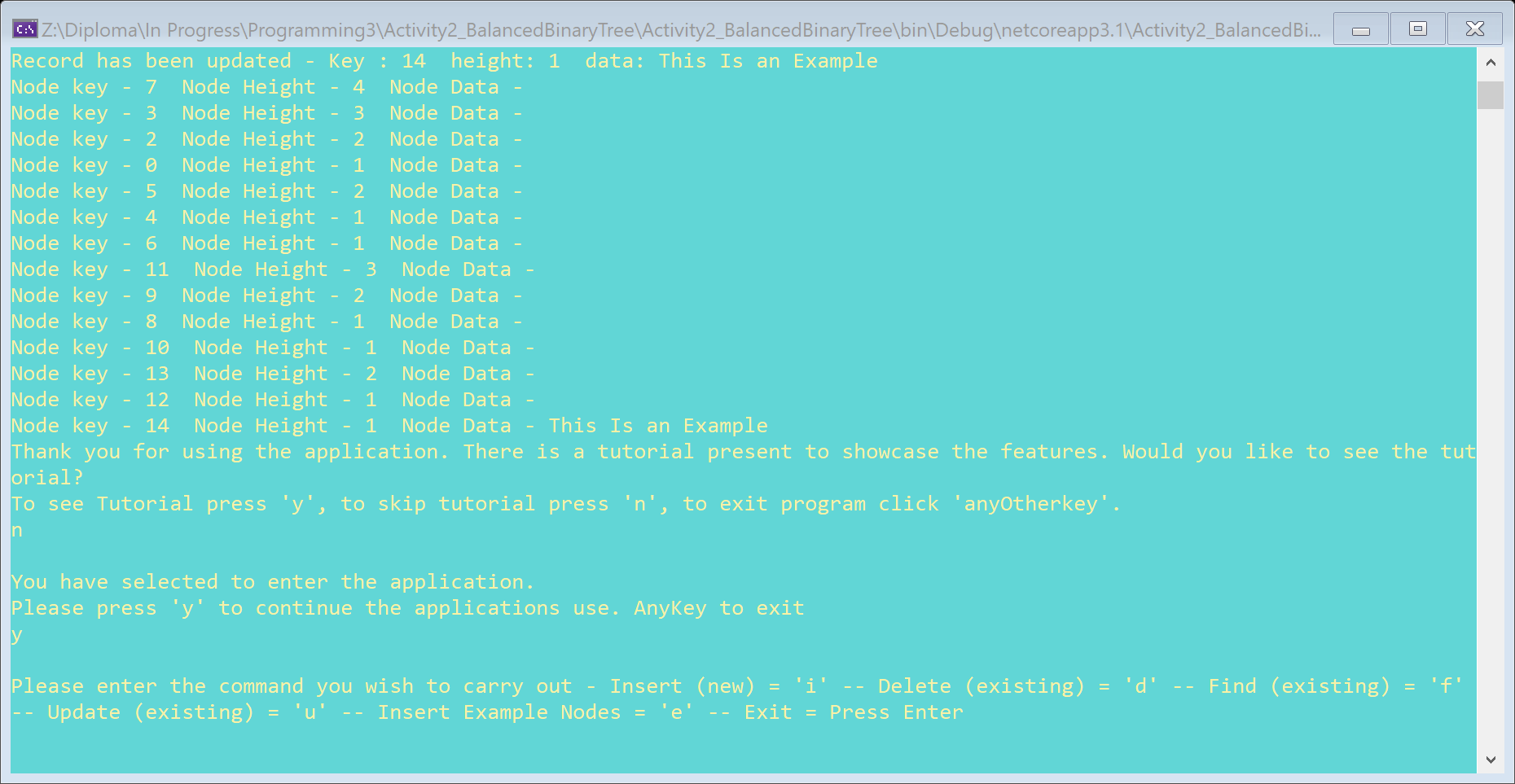


Figure 7. The User can insert individual nodes into the tree using User Input as Node Keys.

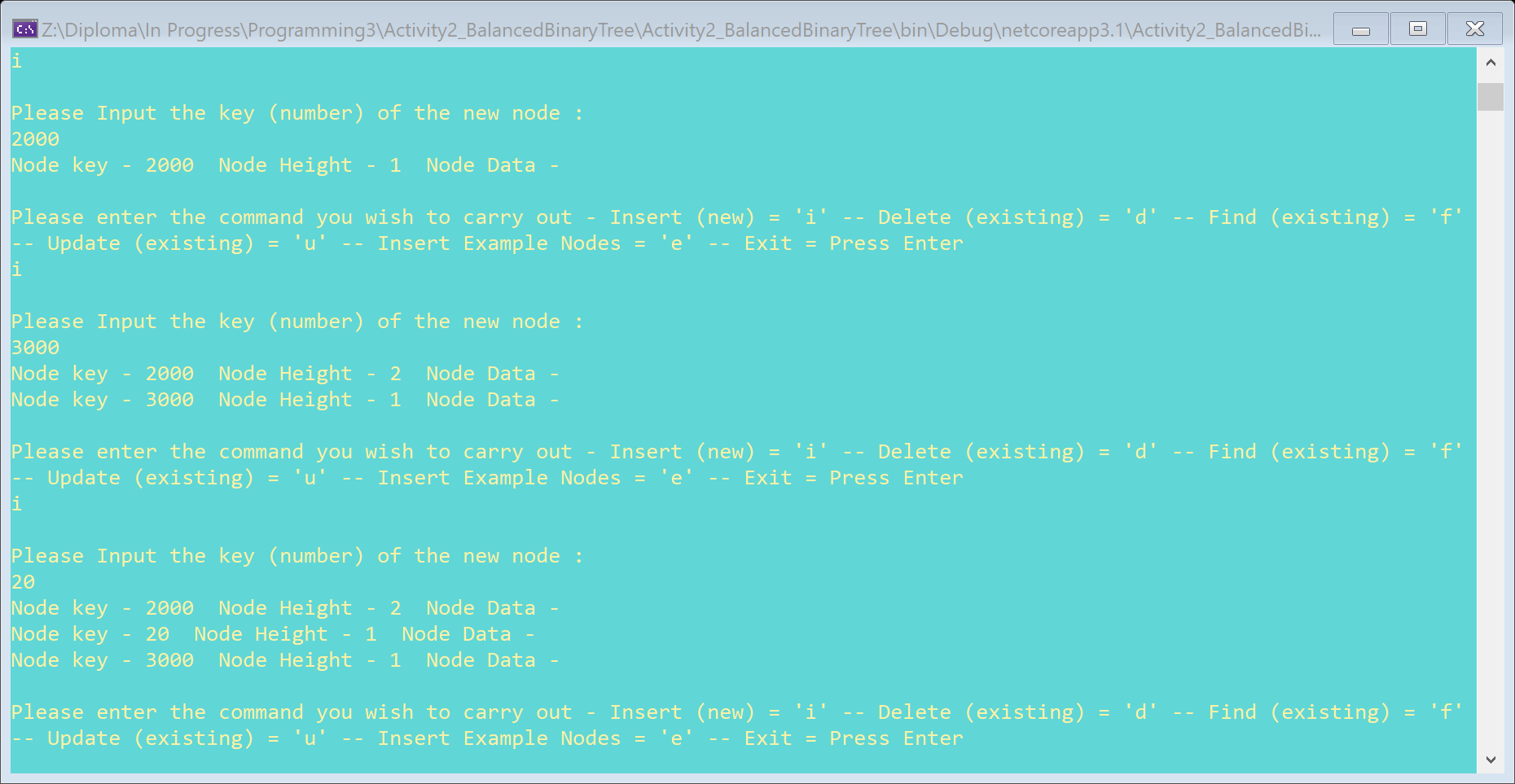


Figure 8. All the methods from the Tutorial (insert, delete, find, update) are available in the looping application.

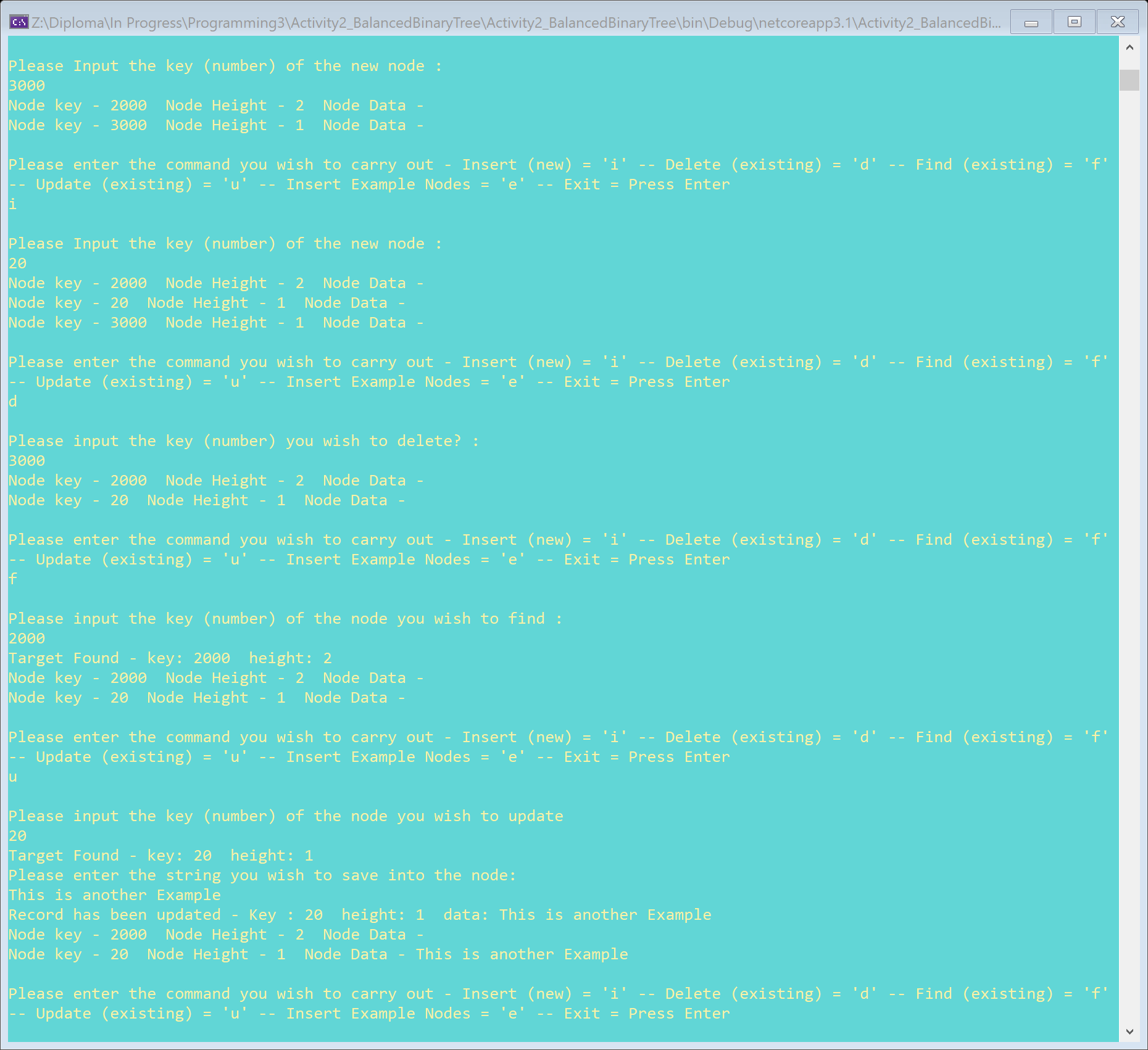
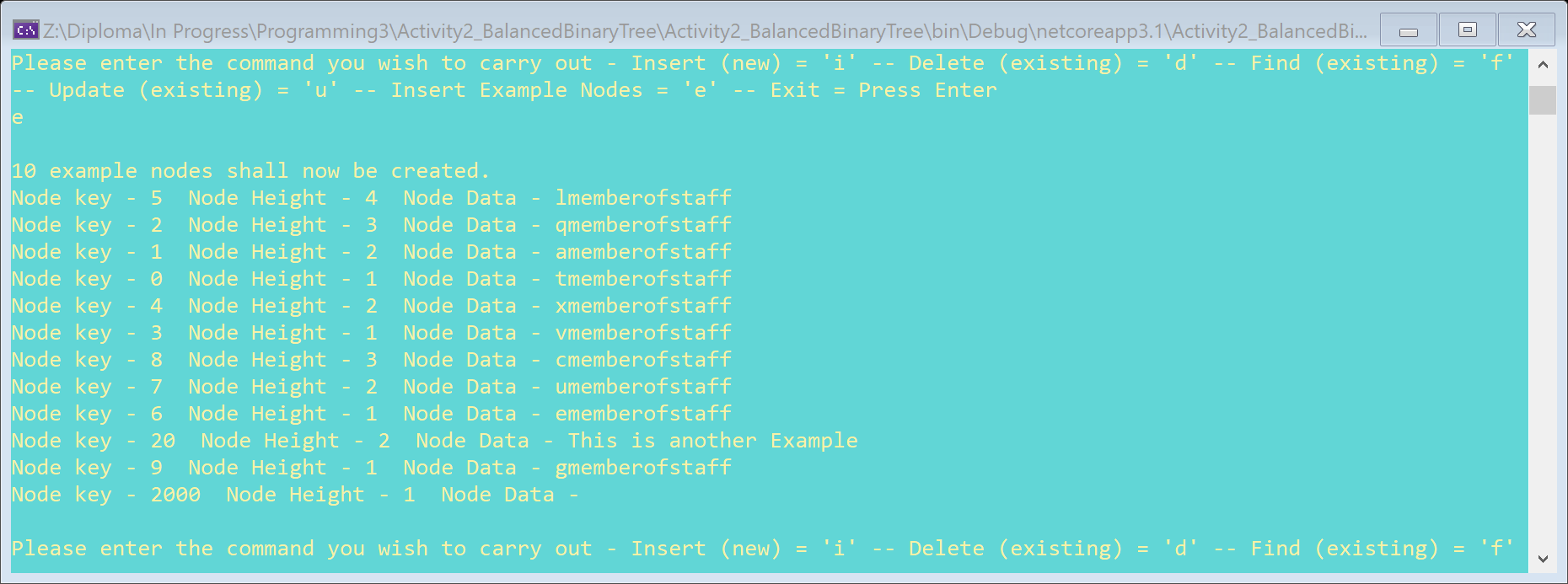


Figure 9. An extra function in the looping application is the option to insert X number of example nodes filled with values. Any keys already existing within the tree has their data replaced with an example data String.



## Testing Tables

All buttons shall be tested for functionality. Play Next and Play Previous shall be used when the adjacent song is there or valued at null. All buttons shall be tested while there are no songs present in the list.

|  |  |  |
| --- | --- | --- |
| **Test ID, Description** | **Prediction** | **Actual, Pass / Fail** |
| 1) Figure 1. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 2) Figure 2. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 3) Figure 3. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 4) Figure 4. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 5) Figure 5. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 6) Figure 6. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 7) Figure 7. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 8) Figure 8. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 9) Figure 9. Follow Onscreen Instruction | Application will continue as normal. | Pass |
| 10) Figure 1. Input ‘n’ option. | Skips Tutorial and Proceeds to looping application. | Pass |
| 11) Figure 1. Press Enter. | Exits Program after Message is printed. | Pass |
| 12) Figure 6. Input ‘n’ option. | Exits Program after Message is printed. | Pass |
| 13) Figure 6. Press Enter. | Exits Program after Message is printed. | Pass |