Isaiah Holder

Lab-8

Array implementation of a BST without using recursion

**Operations Input Values Expected Output**

*InsertItem(int):*

Insert 10 values into the tree. 10,20,23,24,22,5,3,9,7,2,4 void

*Print(PRE\_ORDER,root):*

Print the values usine the

pre-order traversal. void 10,5,3,2,4,9,7,20,23,22,24

*Print(IN\_ORDER,root):*

Print the values using the

in-order traversal. void 2,3,4,5,7,9,10,20,22,23,24

*Print(POST\_ORDER,root):*

Print the values using the

post-order traversal. void 2,4,3,7,9,5,22,24,23,20,10

*RetrieveItem(&bool,int)-Positive Case:*

Search for a value known tobe

included in the original set of

numbers inserted into the structure. 23 Item Found

*RetrieveItem(&bool,int)-Negitive Case:*

Search for a value known tobe

absent from the original set

of numbers inserted into the structure. 230 Item Not Found

*DeleteItem()-No childeren:*

Delete a node with no childeren 7 2,3,4,5,9,10,20,22,23,24

*DeleteItem()-Right child:*

Delete a node with only one child

branching to the right. 20 2,3,4,5,9,10,22,23,24

*DeleteItem()-Two childeren/Left child:*

Delete a node with two childeren. 10 2,3,4,5,9,22,23,24

**Continue onto next page**

***Continued from page one:***

**Operations Input Values Expected Output**

*GetNextItem():*

The GetNextItem function is void void

tested within the print function.

*GetLength():*

Call GetLength void 8

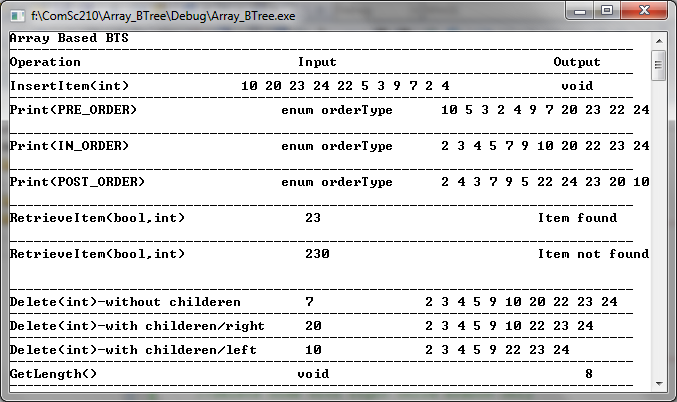
*MakeEmpty():*

Call MakeEmpty() void void

*ResetTree():*

Call ResetTree() void void

***Screen shot of operation:***

******