******

***PROJECT REPORT***

**“Binary Search Tree”**

***COURSE ID: EE-229***

***COURSE NAME: COMPUTER ORGANIZATION AND***

***ASSEMBLY LANGUAGE***

***TEACHER NAME: MUHAMMAD DANISH KHAN***

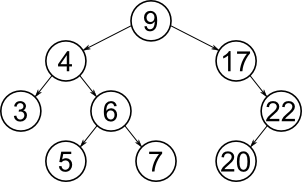
***GROUP MEMBERS:***

***• SHOUKAT ALI* [18K-1257]**

***• MUHAMMAD AWATIF ANSARI* [18K-1155]**

***INTRODUCTION***

***A binary search tree (BST), also known as an ordered binary tree, is a node-based data structure in which each node has no more than two child nodes. Each child must either be a leaf node or the root of another binary search tree. The left sub-tree contains only nodes with keys less than the parent node; the right sub-tree contains only nodes with keys greater than the parent node.***

******

***The BST data structure is the basis for a number of highly efficient sorting and searching algorithms, and it can be used to construct more abstract data structures including sets, multisets, and associative arrays.***

***Programme Interface***

* ***The program is user friendly.***
* ***First of all a data type selection menu will pop up infront of the screen having options of working on Integer data type or Charachter data type.***
* ***There are various choices given to the user to select and perform that specific task.***
* ***The tasks are as follows:***
* ***Insertion in the tree.***

***This function firstly asks the user as how many values the user wishes to add, and then after taking the input from user, it insert the values accordingly into the binary search tree.***

* ***Tree Traversal.***

***This function is basically doing the traversal throughout the tree by the concept of In-Order, which gives the sorted elements of the tree as the output on the console.***

* ***Searching a particular node by the data.***

***This function is basically for searching a element, given by the user as input, in the tree.***

* ***Minimum Value Node.***

***This function finds the minimum value from teh tree by traversing to the extreme left in the tree.***

* ***Maximum Value Node.***

***This function finds the maximum value from teh tree by traversing to the extreme right in the tree.***

* ***Deleteing a node and then managing the tree in order.***

***This function deletes a node in the tree by value, it contains three cases as follows:***

1. ***Deleting the node with no children.***
2. ***Deleting the node with only one child (left or right).***
3. ***Deleting the children with both left and right child.***

* ***End of program.***

***It will end the program as soon as the user selects this choice.***

* ***User can perform every task willingly until he chooses to end the program.***

***Program Execution***

* ***The program is constructed through Visual Studio using Assembly Language x86 CISC.***
* ***In the output program shows the output with respect to the operations performed by user.***

***Input and Output***

* ***The user will face the Choices firstly on the console i.e Menu window.***
* ***User will provide ann input as to select a function.***
* ***User will be seeing the respective function that he chose only on the console, with no other function interruption.***
* ***The menu will repeatedly appears after performing a tasks unless the user chooses to End the program.***

***Program Structure***

***Program contains use of:***

* ***Visual Studio (Assembly Language x86)***
* ***Irvine Library***
* ***Macros***
* ***String Premitives***
* ***Calling Conventions***
* ***Recursion***
* ***Stacks***
* ***Proto, Invoke***
* ***Pointers***

***Improvements and Extensions***

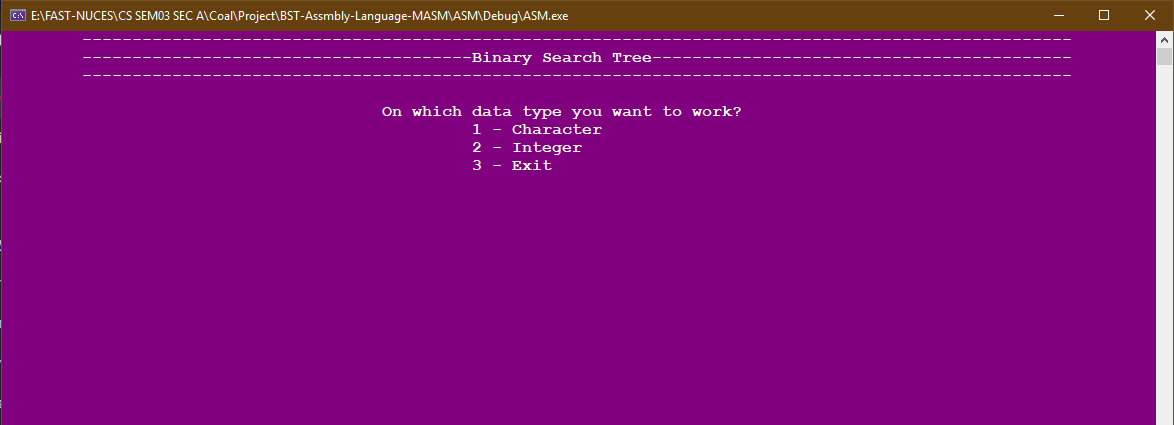
* ***One of the strong point of our program is that a person can implement a BST without even using Inline assembly and with easy to understand coding.***
* ***We can improve our Project by making it more comprehensive from user’s point of view and needs.***
* ***We can enhance our project by modifying it in inline assembly interface using C++ language.***

***Difficulties Encountered***

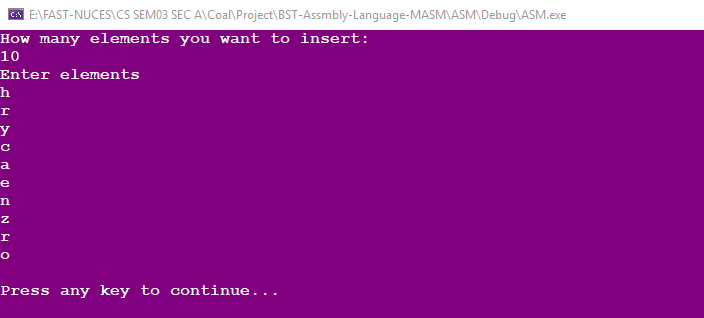
* ***We encountered many problems such as Traversing and deleting the nodes with different cases.***
* ***Understanding the Assembly Language and coding Data Structures in it was challenging and very much new for us.***
* ***One of the major difficulties we faced is Deleting the given node.***
* ***After deleting the node, making the link accordingly with the lower nodes in the tree was a bit challenging.***

***Screen Shots***

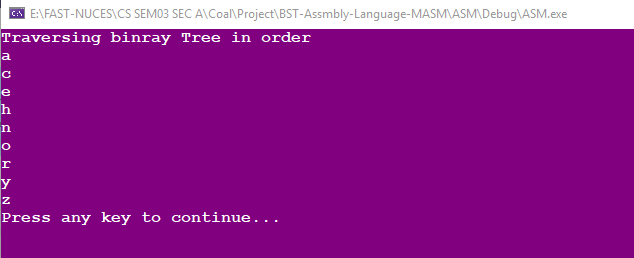
* ***Main Menu***

******

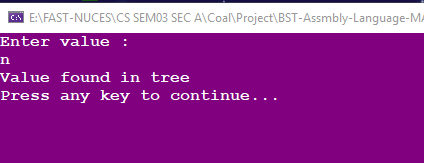
* ***Character Data Type:***
  + ***Insertion:***

******

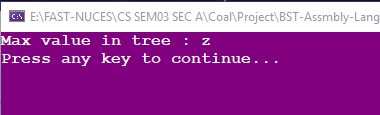
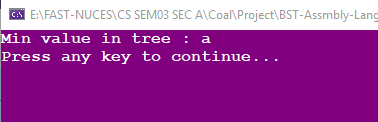
* + ***Traversal:***

******

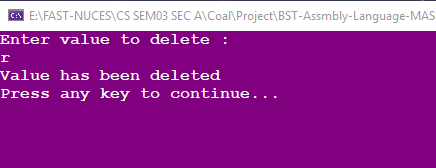
* + ***Searching:***

******

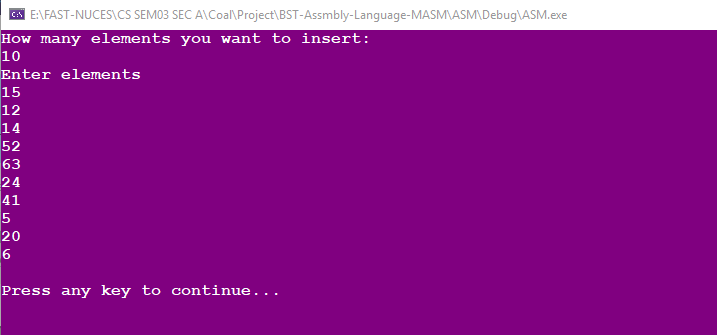
* + ***Minimum and Maximum:***

******

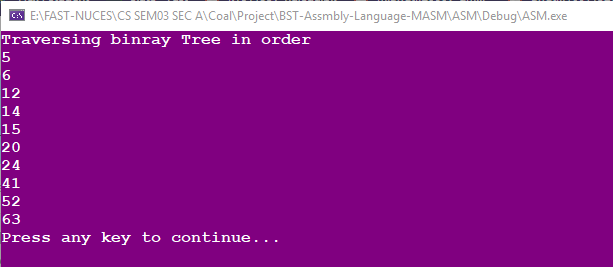
* + ***Deleting:***

******

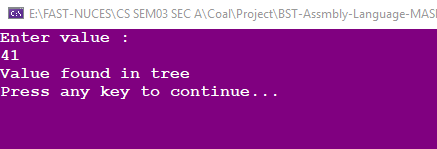
* ***Integer Data Type:***
  + ***Insertion:***

******

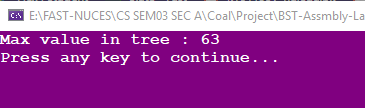
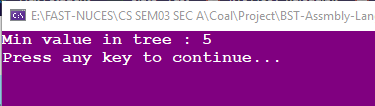
* + ***Traversal:***

******

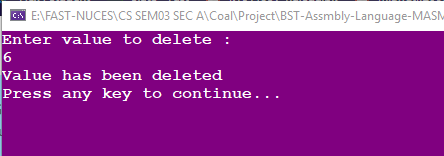
* + ***Searching:***

******

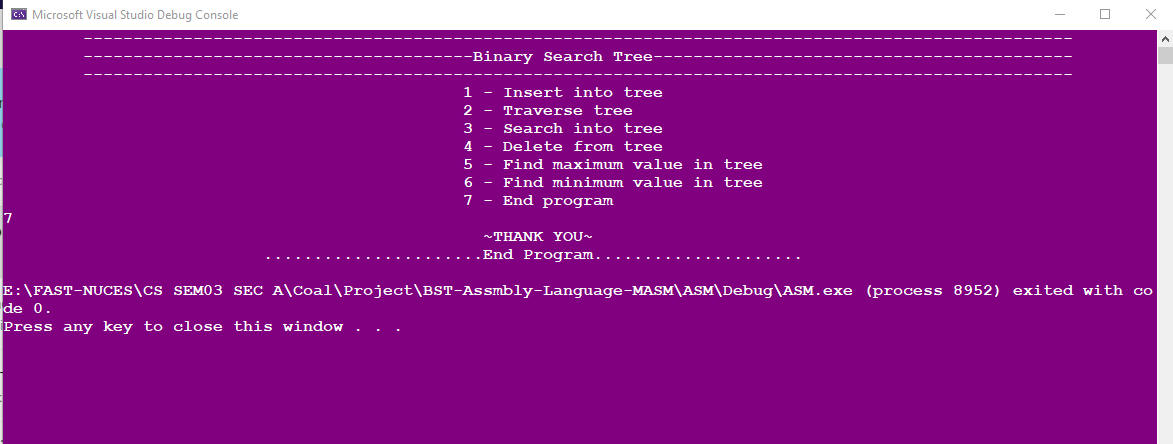
* + ***Minimum and Maximum:***

******

* + ***Deleting:***

******

* ***End Of program***

******

***Conclusion***

***In conclusion, our program is secured, user friendly and easy to use for any novice, in which every user can perform their respective tasks without having any technical issues or further guidance.***