COP 5536 Fall 2022
Programming Project

Name: Arijit Dutta
UFID: 55889097
Email: a.dutta@ufl.edu

Running the Code -

- Unzip the file dutta.arijit.zip
- $make
- ./avltree <input_filename>
- Output will be present in the output_file.txt and also will be printed in the console. The tree is also printed in the console after every operation in a level order format.

Avl.cpp code overview -

- Code Structure:
  - **Struct node** that contains
    - Left pointer (struct node*)
    - Right pointer (struct node*)
    - Data (int)
    - Height (int)
  - **Class AVL**
  - **Methods**
    - findHeight(struct node*):int - Calculates and returns the height of a node.
    - balFactor(struct node*):int - Returns the balance factor of the node [height of left subtree - height of right subtree].
    - llRotation(struct node*):struct node* - Takes a node and performs a LL Rotation with respect to that node.
    - lrRotation(struct node*):struct node* - Takes a node and performs a LR Rotation with respect to that node.
    - rrRotation(struct node*):struct node* - Takes a node and performs a RR Rotation with respect to that node.

        - rlRotation(struct node*):struct node* - Takes a node and performs a RL Rotation with respect to that node.
        - insertNode(struct node*, int):struct node* - Takes the root of the tree and a key value and inserts a new node with the given key value in the tree.
        - deleteNode(struct node*, int):struct node* - Takes the root of the tree and a key value. The method deletes the node having the input key value.
        - inorderPred(struct node*):struct node* - Gives the inorder predecessor of the input node. It gives the rightmost child of the left subtree of the input node.
        - inorderSucc(struct node*):struct node* - Gives the inorder successor of the input node. It gives the leftmost child of the right subtree of the input node.
        - printTree():void - Prints the tree in a level order format.
        - searchNode(struct node*, int):struct node* - This methodTakes the root and a search key. Returns the node with data equal to the search key or returns null if there is no such node.
        - searchNode(struct node*, int, int, vector<struct node*>):void - Gives the nodes whose value falls within the given input range.
    - **main(int, char**):int** - The main function where the input file is read, operations are decoded from the input, their respective methods are called and the outputs written in the output file.
        - searchTwoKey(int, int, AVL *, std::ofstream&):void - This method is called from main if the operation is to search keys within a range.
        - operations(char*, int, int, AVL *, std::ofstream&):void - This method is called from main if the operations are insert, delete or search a key.