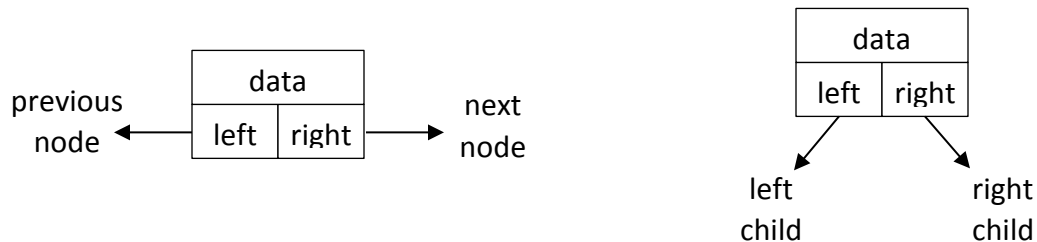


Write a C++ program project4.cpp that operates as described below.

First observe that the nodes of a circular doubly-linked list and a binary search tree both have the same essential structure, that is, a data field and two links to other nodes.



Your program will contain a class ListTree that at different times may represent either a sorted circular doubly-linked list or a binary search tree. Here is some possible code to get you started. You may borrow and adapt this code as needed.

```
struct Node {
    int data;
    Node *left, *right;
    ...
};

class ListTree {
    bool isTree;
    Node *start;    // either the root of tree or front of list
public:
    ListTree( ) {
        isTree = true;    // initially it is a tree
        start = NULL;
    }
    void insert (int x) {
        if (isTree) treeInsert(x, start);
        else listInsert(x, start);
    }
    void display( ) {
        if (isTree) treeDisplay (start);
        else listDisplay (start);
    }
    void toggle( ) {
        if (isTree) treeToList (start);
        else listToTree (start);
        isTree = ! isTree;
    }
    ...
};
```

Your program will read these input commands from standard input until end-of-input occurs.

insert X	Insert the integer value X into the ListTree.
display	Display the contents of the ListTree using the formats shown in the example below.
toggle	Convert between the list representation and the tree representation. If the ListTree is currently a list, then it becomes a tree, or vice versa. Each conversion must be performed in-place; do not allocate any new nodes during the conversion. Finally, when converting to a tree, the resulting tree and each of its subtrees must be balanced, with the middle node at the root, and half the nodes in the left subtree, as shown in the example below. Hint: use recursion.

Each of the above operations should run in  $O(n)$  time, where  $n$  is the current number of nodes. Your program will write the results of each display command to standard output. Example:

Input:	Output:
insert 1	
display	(1)
insert 5	
display	(1 (5))
insert 0	
display	((0) 1 (5))
insert 7	
insert 3	
display	((0) 1 ((3) 5 (7)))
insert 8	
insert 2	
display	((0) 1 (((2) 3) 5 (7 (8))))
insert 6	
insert 4	
display	((0) 1 (((2) 3 (4)) 5 ((6) 7 (8))))
toggle	
display	[0,1,2,3,4,5,6,7,8]
toggle	
display	((((0) 1) 2 (3)) 4 (((5) 6) 7 (8)))
insert 11	
insert 9	
insert 13	
display	(((((0) 1) 2 (3)) 4 (((5) 6) 7 (8 ((9) 11 (13))))))
toggle	
display	[0,1,2,3,4,5,6,7,8,9,11,13]
insert 10	
insert 12	
display	[0,1,2,3,4,5,6,7,8,9,10,11,12,13]
toggle	
display	(((((0) 1 (2)) 3 ((4) 5 (6))) 7 (((8) 9 (10)) 11 ((12) 13)))

The blank lines shown in the previous example output are not required. They are just there to help show which display command produces each line of output.

Please carefully read the following requirements:

- You must do your own work. You must not borrow any code from any other person, book, website, or any other source. You also must not share your code with any other person, or post it on any website. We run plagiarism detection software on every project. So if you violate these rules, you may receive an invitation to the dean's office to discuss the penalties for academic misconduct.
- Because a purpose of this course is to learn how to properly implement data structures and algorithms, it is not permitted in general to include STL libraries such as `<list>`, `<vector>`, `<stack>`, `<queue>`, `<deque>`, `<algorithm>`, `<numeric>`, `<utility>`, ... in your program. You are always permitted to use `<iostream>`, `<string>`, and any C libraries such as `<cstdlib>` and `<cmath>`. But do not include any other libraries unless explicitly allowed on the assignment.
- Make sure your program runs properly on the cs-intro.ua.edu server, because this is where your program will be graded. In particular, make sure your program initializes the values of all variables when they are declared or allocated. Otherwise it might behave differently on Linux than it does on a PC or Mac.
- Your program will be compiled using this command: `g++ project4.cpp -Wall -lm -std=c++11`. Alternatively, if you split your program into multiple files, then it will instead be compiled using this command: `g++ *.cpp -Wall -lm -std=c++11`.
- Verify that all the necessary .h and .cpp files which are needed to compile your program are included in the same directory before you submit your program. There should be no extra subdirectories and no extra .h or .cpp files, otherwise your program might not compile the way you intended.
- Compress your project into a zip file that contains your C++ program source file. Right-click (or secondary click) on your project directory, and then (depending on your operating system) select either the Compress option or Send To → Compress from the popup menu. Finally upload your .zip file that contains your .cpp file for this project to Blackboard.
- If you violate the above requirements such that it breaks our grading script, your project will be assessed a significant point deduction, and extreme or multiple violations may cause the project to be considered ungradable.
- Every semester many students lose some points because they don't follow all the instructions. So please read and follow all the project specifications precisely to prevent losing points unnecessarily. If anything is unclear, please ask for clarification well before the project is due. Please pay particular attention to input and output formats.
- Submit your project on Blackboard by the due date (11:59pm Friday). There is a grace period of 24 hours (until 11:59pm Saturday). Projects submitted on Sunday will be assessed a late penalty of 5% per hour. No projects will be accepted after Sunday. Once it is graded, your project score will be posted on Blackboard and the results of the grading script will be sent to your Crimson email account.
- Double-check and triple-check your submission when you submit it. Errors discovered later cannot be fixed and resubmitted after the project is graded. Projects will not be re-graded unless an error is found in the grading script or in the input/output files that are used during grading.