TreeIsPerfectlyBalanced

Your task is to write a function, TreeIsPerfectlyBalanced, that determines whether a given tree is perfectly balanced. A tree is perfectly balanced if, for every node, the difference in size between its left and right subtrees does not exceed 1. The function should return true if the tree is perfectly balanced, and false otherwise.

Download

Click here to download a zip of the files.

The Files

Tree.c	Contains code for reading and printing a binary tree
Tree.h	Contains the definition of the binary tree data structure and function prototypes
testTreeIsPerfectlyBalanced.c	Contains the main function, which reads in a binary tree from standard input, calls TreeIsPerfectlyBalanced, and prints out the result.
TreeIsPerfectlyBalanced.c	Contains TreeIsPerfectlyBalanced, the function you must implement
Makefile	A makefile to compile your code
tests/	A directory containing the inputs and expected outputs for some basic tests
autotest	A script that uses the tests in the tests directory to autotest your solution. You should only run this after you have tested your solution manually.

Examples

Your program should behave like these examples:

```
$ ./testTreeIsPerfectlyBalanced
Enter the preorder traversal of the tree: 3 2 1 4 5
Enter the in-order traversal of the tree: 1 2 3 4 5
Tree:

    3
   / \
  2   4
 /     \
1       5


TreeIsPerfectlyBalanced returned: TRUE
```

```
$ ./testTreeIsPerfectlyBalanced
Enter the preorder traversal of the tree: 1 2 4 7 5 3 6
Enter the in-order traversal of the tree: 7 4 2 5 1 6 3
Tree:

        1
       / \
      /   \
     /     \
    2       3
   / \     /
  4   5   6
 /
7


TreeIsPerfectlyBalanced returned: FALSE
```

```
$ ./testTreeIsPerfectlyBalanced
Enter the preorder traversal of the tree:
Enter the in-order traversal of the tree:
Tree:

X

TreeIsPerfectlyBalanced returned: TRUE
```

Testing

You can test your program manually by compiling your code using `make`, and then running `./testTreeIsPerfectlyBalanced`, as shown above. After you are satisfied with your solution, you can autotest it by running `./autotest`. This will run some basic tests on your program.