TreeHeight

Your task is to write a function, TreeHeight, that returns the height of the given tree. The height of a tree is the number of edges on the longest path from the root node to a leaf node. The height of an empty tree is considered to be -1.

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
testTreeHeight.c	Contains the main function, which reads in a binary tree from standard input, calls TreeHeight, and prints out the result.
TreeHeight.c	Contains TreeHeight, 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:

```
$ ./testTreeHeight
Enter the preorder traversal of the tree: 2 5 7 9 6
Enter the in-order traversal of the tree: 5 2 9 7 6
Tree:

  2
 / \
5   7
   / \
  9   6

TreeHeight returned: 2

Explanation: The given tree does not necessarily have to be a BST.
```

```
$ ./testTreeHeight
Enter the preorder traversal of the tree: 1 8 3 2 6 4 5 7 9
Enter the in-order traversal of the tree: 1 2 3 4 5 6 7 8 9
Tree:

  1
   \
    8
   / \
  3   9
 / \
2   6
   / \
  4   7
   \
    5

TreeHeight returned: 5
```

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

X

TreeHeight returned: -1
```

```
$ ./testTreeHeight
Enter the preorder traversal of the tree: 5 8 3 9 1 7 4
Enter the in-order traversal of the tree: 1 9 3 8 5 7 4
Tree:

        5
       / \
      8   7
     /     \
    3       4
   /
  9
 /
1

TreeHeight returned: 4
```

Testing

You can test your program manually by compiling your code using `make`, and then running `./testTreeHeight`, 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.