BSTreeNodeDepth

Your task is to write a function, BSTreeNodeDepth, that returns the depth of the node containing the given key in the tree if it exists, or -1 otherwise.

Download

Click here to download a zip of the files.

The Files

BSTree.c	Contains code for reading and printing a BST
BSTree.h	Contains the definition of the BST data structure and function prototypes
testBSTreeNodeDepth.c	Contains the main function, which reads in a BST from standard input, calls BSTreeNodeDepth for each key read in, and prints out the results.
BSTreeNodeDepth.c	Contains BSTreeNodeDepth, 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:

```
$ ./testBSTreeNodeDepth
./testBSTreeNodeDepth
Enter the preorder traversal of the BST: 9 4 2 7
Tree:

    9
   /
  4
 / \
2   7

Enter key: 1
For key = 1, BSTreeNodeDepth returned -1
Enter key: 2
For key = 2, BSTreeNodeDepth returned 2
Enter key: 3
For key = 3, BSTreeNodeDepth returned -1
Enter key: 4
For key = 4, BSTreeNodeDepth returned 1
Enter key: 5
For key = 5, BSTreeNodeDepth returned -1
Enter key: 6
For key = 6, BSTreeNodeDepth returned -1
Enter key: 7
For key = 7, BSTreeNodeDepth returned 2
Enter key: 8
For key = 8, BSTreeNodeDepth returned -1
Enter key: 9
For key = 9, BSTreeNodeDepth returned 0
Enter key: 10
For key = 10, BSTreeNodeDepth returned -1
Enter key: (Ctrl + D)
```

```
$ ./testBSTreeNodeDepth
Enter the preorder traversal of the BST: 1 8 3 6
Tree:

1
 \
  8
 /
3
 \
  6

Enter key: 0
For key = 0, BSTreeNodeDepth returned -1
Enter key: 1
For key = 1, BSTreeNodeDepth returned 0
Enter key: 2
For key = 2, BSTreeNodeDepth returned -1
Enter key: 3
For key = 3, BSTreeNodeDepth returned 2
Enter key: 4
For key = 4, BSTreeNodeDepth returned -1
Enter key: 5
For key = 5, BSTreeNodeDepth returned -1
Enter key: 6
For key = 6, BSTreeNodeDepth returned 3
Enter key: 7
For key = 7, BSTreeNodeDepth returned -1
Enter key: 8
For key = 8, BSTreeNodeDepth returned 1
Enter key: 9
For key = 9, BSTreeNodeDepth returned -1
Enter key: (Ctrl + D)
```

Testing

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