BSTreeGetSmallest

Your task is to write a function, BSTreeGetSmallest, that returns a pointer to the node containing the smallest value in the given BST. If the tree is empty, return NULL.

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
testBSTreeGetSmallest.c	Contains the main function, which reads in a BST from standard input, calls BSTreeGetSmallest, and prints out the result.
BSTreeGetSmallest.c	Contains BSTreeGetSmallest, 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:

```
$ ./testBSTreeGetSmallest
Enter the preorder traversal of the BST: 6 5 2 8 9

Tree:

    6
   / \
  5   8
 /     \
2       9

BSTreeGetSmallest returned: 2
```

```
$ ./testBSTreeGetSmallest
Enter the preorder traversal of the BST: 5 8 6 9

Tree:

5
 \
  8
 / \
6   9

BSTreeGetSmallest returned: 5
```

```
$ ./testBSTreeGetSmallest
Enter the preorder traversal of the BST:

Tree:

X

BSTreeGetSmallest returned: NULL
```

Testing

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