BSTreePostfix

Your task is to write a function, BSTreePostfix, that prints out the values of the given BST in postfix order. The values should be printed out space-separated on a single line.

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
testBSTreePostfix.c	Contains the main function, which reads in a BST from standard input and calls BSTreePostfix.
BSTreePostfix.c	Contains BSTreePostfix, 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:

```
$ ./testBSTreePostfix
Enter the preorder traversal of the BST:
Tree:

X

Values in postfix order:
```

```
$ ./testBSTreePostfix
Enter the preorder traversal of the BST: 7 4 3 1 10 8 13
Tree:

       7
      / \
     /   \
    4    10
   /     / \
  3     8  13
 /
1

Values in postfix order: 1 3 4 8 13 10 7
```

```
$ ./testBSTreePostfix
Enter the preorder traversal of the BST: 8 7 4 2 1 3 6
Tree:

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

Values in postfix order: 1 3 2 6 4 7 8
```

```
$ ./testBSTreePostfix
Enter the preorder traversal of the BST: 6
Tree:

6

Values in postfix order: 6
```

Testing

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