Binary Search Trees

INSTRUCTIONS

Use the following definition for a BTNode, as provided in the lecture slides:

```
class BTNode:
    def __init__(self, item, left=None, right=None):
        self.item = item  # Store the item (integer)
        self.left = left  # Reference to the left child node
        self.right = right  # Reference to the right child node
```

----

1. Write a function `insertBSTNode()` that adds an item to a Binary Search Tree.

   ```
   def insertBSTNode(node, value):
   ```

   BST nodes should be dynamically created by instantiating the `BTNode class`.

   Hint: The core logic of this function has been covered in the lecture slides. Ensure that your implementation correctly handles inserting a node into an empty BST.

2. Write a function `printBSTInOrder()` that prints the items stored in a Binary Search Tree in sorted order using an in-order traversal pattern.

   ```
   def printBSTInOrder(node):
   ```

3. Write a function `isBST()` that determines whether a given Binary Tree is also a Binary Search Tree. The function should return True if the tree is a BST and False otherwise.

   ```
   def isBST(node):
   ```

4. Write a function `removeBSTNode()` that removes a given item from a Binary Search Tree. The function should return 0 if the item was successfully removed and -1 otherwise.

   ```
   def removeBSTNode(node, value):
   ```