Convert Binary Search Tree to Double Linked list

```
#include <stdio.h>
#include <stdlib.h>

struct node
{
	int data;
	struct node *left;
	struct node *right;
};

struct node *newNode(int data)
{
	struct node *temp = (struct node *)malloc(sizeof(struct node));
	temp->data = data;
	temp->left = temp->right = NULL;
	return temp;
}

struct node *convertBSTtoDLL(struct node *root)
{
	if(root->left)
	{
		struct node *lTree = convertBSTtoDLL(root->left);
		for(;lTree->right; lTree = lTree->right); //inorder predecesor
		lTree->right = root;
		root->left = lTree;
	}
	if(root->right)
	{
		struct node *rTree = convertBSTtoDLL(root->right);
		for(;rTree->left; rTree = rTree->left); //inorder successor
		rTree->left = root;
		root->right = rTree;
	}
	return root;
}

void printDLL(struct node *head)
{
	struct node *temp = head;
	while(temp)
	{
		printf("%d\t", temp->data);
		temp = temp->left;
	}
}

int main()
{
```

```
        struct node *root, *lca, *head;
        root = newNode(25);
        root->left = newNode(10);
        root->right = newNode(30);
        root->left->left = newNode(5);
        root->left->right = newNode(15);
        root->left->right->left = newNode(12);
        head = convertBSTtoDLL(root);
        printDLL(head);
        return 0;
}
```

Time complexity: O(n)
Space Complexity: O(1)