Inorder Traversal without using Stack

```
#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;
}

void printinorderWithoutStack(struct node *root)
{
	struct node *currNode, *preNode;
	currNode = root;
	while(currNode)
	{
		if(!currNode->left)
		{
			printf("%d\t", currNode->data);
			currNode = currNode->right;
		}
		else
		{
			preNode = currNode->left;
			while(preNode->right && preNode->right!=currNode)
				preNode = preNode->right;
			if(!preNode->right)
			{
				preNode->right = currNode;
				currNode = currNode->left;
			}
			else
			{
				printf("%d\t", currNode->data);
				preNode->right = NULL;
				currNode = currNode->right;
			}
		}
	}
}

int main()
```

```
{
        struct node *root = newNode(10);
        root->left = newNode(8);
        root->right = newNode(2);
        root->left->left = newNode(3);
        root->left->right = newNode(5);
        root->right->right = newNode(2);
        printinorderWithoutStack(root);
        return 0;
}
```

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