BST level order

```
#include <stdio.h>
#include <stdlib.h>
//level order
struct node
{
	int data;
	struct node* left;
	struct node* right;
};
struct node* queue[100] = {NULL};
struct node* insert(struct node* root,int x)
{
	if(root == NULL)
	{
		struct node* temp = (node*)malloc(sizeof(struct node));
		temp->data = x;
		temp->left = NULL;
		temp->right = NULL;
		root = temp;
		return root;
	}
	else if(x <= root->data)
		root->left = insert(root->left,x);
	else
		root->right = insert(root->right,x);
	return root;
}
void levelorder(struct node* root)
{
	if(root == NULL) return;
	int start=0,end=0;
	queue[end] = root;
	end++;
	while(start != end)
	{
		struct node* curr = queue[start];
		printf("%d ",curr->data);
		if(curr->left != NULL)
		{
			queue[end] = curr->left;
			end++;
		}
		if(curr->right != NULL)
		{
			queue[end] = curr->right;
			end++;
		}
```

```
                start--;
        }
}
int main()
{
        struct node* root = NULL;
        int n,x;
        while(1)
        {
                scanf("%d",&n);
                if(n == 1)
                {
                        scanf("%d",&x);
                        root = insert(root,x);
                }
                if(n == 2)
                {
                        //level order
                        levelorder(root);
                }
                if(n == 3)
                {
                        break;
                }
        }
        return 0;
}
```