```
#include<stdio.h>
#include<iostream>
#include<stdlib.h>
#include<stack>
#include<queue>

using namespace std;

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

struct BinaryNode * createBinaryNode(int value)
{
    struct BinaryNode *B=(struct BinaryNode *)malloc(sizeof(struct BinaryNode));
    B->left=NULL;
    B->right=NULL;
    B->data=value;
    return B;
};
queue<BinaryNode *> q;
stack<BinaryNode *> s;

void  zigZagTraversal(struct BinaryNode *root)
{
    int L2R=1;
    if(!root)
        return;
    q.push(root);
    q.push(NULL);
    while(!q.empty())
    {
        struct BinaryNode * temp=q.front();
        q.pop();
        if(temp==NULL)
        {
            if(L2R==0)
            {
                while(!s.empty())
                {
                    struct BinaryNode *temp1=s.top();
                    printf("%d ",temp1->data);
                    s.pop();
                }
            }
            if(!q.empty())
                q.push(NULL);
            else
                EXIT_SUCCESS;

            L2R=1-L2R;



        }
        else
        {
            if(temp->left)
            q.push(temp->left);
            if(temp->right);
            q.push(temp->right);
            if(L2R==1)
                printf("%d ",temp->data);
```

```
            else
                s.push(temp);

        }

}
}

int main()
{
    struct BinaryNode *root=createBinaryNode(10);
    root->right=createBinaryNode(30);
    root->left=createBinaryNode(20);
    root->left->left=createBinaryNode(40);
    root->left->right=createBinaryNode(50);
    root->right->right=createBinaryNode(70);
    root->right->left=createBinaryNode(60);
    root->left->left->left=createBinaryNode(80);
    root->left->left->right=createBinaryNode(90);
    root->left->right->left=createBinaryNode(100);

    zigZagTraversal(root);


}
```