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

int levelHavingMaxSum(struct BinaryNode *root)
{
    int current_sum=0,max_sum=0,ans;
    if(!root)
        return 0;
    q.push(root);
    q.push(NULL);
    int level=1;
    while(!q.empty())
    {
        struct BinaryNode * temp=q.front();
        q.pop();
        if(temp==NULL)
        {
            if(!q.empty())
            {
                q.push(NULL);
            }

            if(current_sum>max_sum)
            {
                max_sum=current_sum;
                ans=level;
            }

            level++;
            current_sum=0;
        }

        else
        {
            current_sum=current_sum+(temp->data);
            if(temp->left)
                q.push(temp->left);
            if(temp->right)
                q.push(temp->right);
        }
    }
    return ans;
}
```

```
int main()
{
    struct BinaryNode *root=createBinaryNode(10);
    root->right=createBinaryNode(2000);
    root->left=createBinaryNode(30);
    root->left->left=createBinaryNode(0);
    root->left->right=createBinaryNode(0);
    root->right->right=createBinaryNode(2031);
    root->right->left=createBinaryNode(0);

    printf("level Having Max Sum is %d",levelHavingMaxSum(root));
}
```