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

int idx = 0;

int get_size(int n)
{
    int l = 1;
    int ct = 1;
    while (l < n)
    {
        l = l + (int)pow(2, ct);
        ct += 1;
    }
    return l;
}

void print_tree(int *tree, int size)
{
    for (int i = 0; i < size; i++)
    {
        printf("%d ", tree[i]);
    }
}

void get_subarray(int *arr, int *ret, int start, int end)
{
    int k = 0;
    for (int i = start; i < end; i++)
    {
        ret[k] = arr[i];
        k += 1;
    }
}

void pre_order(int *tree, int len, int i, int size, int *pre_tree)
{
    int left_node = -1;
    int right_node = -1;
```

```
    if ((i * 2 + 1) < len)
    {
        left_node = tree[i * 2 + 1];
    }
    if ((i * 2 + 2) < len)
    {
        right_node = tree[i * 2 + 2];
    }
    pre_tree[idx++] = tree[i];
    if (left_node != -1)
    {
        pre_order(tree, len, (i * 2 + 1), size, pre_tree);
    }

    if (right_node != -1)
    {
        pre_order(tree, len, (i * 2 + 2), size, pre_tree);
    }
}


void check_complete(int *tree, int size)
{
    int flag=0;
    for (int i = 0; i < size; i++)
    {
        if (tree[i] != -1 && 2 * (i + 1) < size)
        {
            int ln = tree[2 * (i + 1) - 1];
            int rn = tree[2 * (i + 1)];
            if((ln==0 && rn!=0) || (ln!=0 && rn==0))
            {
                flag=1;
            }
        }
    }
    if(flag==1)
    {
        printf("\n\nGiven tree is not a complete tree");
    }
```

```c
    else
        printf("\n\nGiven tree is a complete tree");
}

int main()
{
    int len = 7;
    int tree[7] = {40, 30, 50, 0, 34, 42, 55};
    printf("Entered tree where 0 represents NULL - \n");
    print_tree(tree, len);
    int *pre_tree;
    int size = get_size(len);
    pre_tree = (int *)malloc(size * sizeof(int));
    pre_order(tree, len, 0, size, pre_tree);
    printf("\nPre order traversal : - \n");
    print_tree(pre_tree, size);
    check_complete(pre_tree,size);

    return 0;
}
```

```
PS D:\CS Lab\c code> cd "d:\CS Lab\c code\" ; if ($?) { gcc trial2.c -o trial2 } ; if ($?) { .\trial2 }
Entered tree where 0 represents NULL -
40 30 50 0 34 42 55
Pre order traversal : -
40 30 0 34 50 42 55

Given tree is not a complete tree
PS D:\CS Lab\c code>
```