

Assume each node in the binary search tree contains numeric values as their data field and it is represented using array data structure. Write a 'C' program to find the number of nodes with even number in the left subtree of root node from the binary search tree. Print -1 if there is no node found in the left subtree. Apply tree traversal to visit the nodes.

```
#include <stdio.h>
int tree[10];
int c=0;
int rootnode(int key){
    if(tree[0] != '\0')
        printf("Tree already had root");
    else
        tree[0] = key;
    return 0;
}
int leftchild(int key, int parent){
    if(tree[parent] == '\0')
        printf("Can't set child at %d, no parent found\n", (parent * 2) + 1);
    else
        tree[(parent * 2) + 1] = key;
    return 0;
}
int rightchild(int key, int parent){
    if(tree[parent] == '\0')
        printf("Can't set child at %d, no parent found\n", (parent * 2) + 2);
    else
        tree[(parent * 2) + 2] = key;
    return 0;
}
void InOrder(int arr[], int start, int end)
{
    if(start > end)
        return;
    // print left subtree
    InOrder(arr, start*2 + 1, end);

    //print node
    printf("%d  ", arr[start]);
    if((arr[start]%2)==0){
```

```

        c++;
    }
    // print right subtree
    InOrder(arr, start*2 + 2, end);
}

int main(){
    rootnode(100);
    rightchild(120, 0);
    leftchild(80, 0);
    rightchild(90, 1);
    rightchild(130, 2);
    leftchild(70, 1);
    leftchild(110, 2);
    leftchild(65, 3);
    printf("Inorder BST: ");
    InOrder(tree,0,7);
    printf("\nnumber of even numbers in left subtree\n");
    if(c==0){
        printf("-1");
    }
    else
        printf("%d",c);
}

```

```

PS D:\CS Lab\c code> cd "d:\CS Lab\c code\" ; if ($?) { gcc trial.c -o trial } ; if ($?) { .\trial }
Inorder BST: 65 70 80 90 100 110 120 130
number of even numbers in left subtree
7

```