

Print Root to Leaf Paths in a binary tree

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 1000

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

struct node *newNode(int data)
{
    struct node *temp = (struct node *)malloc(sizeof(struct node));
    temp->data = data;
    temp->left = temp->right = NULL;
    return temp;
}

void printPath(int path[], int index)
{
    for(int i=0; i<index; i++)
        printf("%d\t", path[i]);
    printf("\n");
}

void getRootToLeafPaths(struct node *root, int path[], int index)
{
    if(root)
    {
        path[index++] = root->data;
        if(!(root->left) && !(root->right))
            printPath(path, index);
        else
        {
            getRootToLeafPaths(root->left, path, index);
            getRootToLeafPaths(root->right, path, index);
        }
    }
}

int main()
{
    struct node *root=NULL;
    int path[MAX];
    root = newNode(10);
    root->left = newNode(20);
    root->right = newNode(30);
    root->left->left = newNode(40);
    root->right->left = newNode(50);
}
```

```
    root->right->right = newNode(60);  
    getRootToLeafPaths(root, path, 0);  
    return 0;  
}
```

Time complexity: $O(n)$

Space Complexity: $O(n)$