**BINARY TREE—traversing depth----**

27, 14, 35, 10, 19, 31, 42



| NULL | 27 | NULL |
| --- | --- | --- |

27, 14, 35, 10, 19, 31, 42



27, 14, 35, 10, 19, 31, 42









27, 14, 35, 10, 19, 31, 42











27, 14, 35, 10, 19, 31, 42











27, 14, 35, 10, 19, 31, 42











27, 14, 35, 10, 19, 31, 42











**node \*create()**

**{**

**node \*p;**

**int x;**

**printf("Enter data(-1 for no data):");**

**scanf("%d",&x);**

**if(x==-1)**

**return NULL;**

**p=(node\*)malloc(sizeof(node));**

**p->data=x;**

**printf("Enter left child of %d:\n",x);**

**p->left=create();**

**printf("Enter right child of %d:\n",x);**

**p->right=create();**

**return p;**

**}**

**BINARY TREE----LEVEL ORDER**

27, 14, 35, 10, 19, 31, 42



27, 14, 35, 10, 19, 31, 42



27, 14, 35, 10, 19, 31, 42





27, 14, 35, 10, 19, 31, 42









27, 14, 35, 10, 19, 31, 42











27, 14, 35, 10, 19, 31, 42











27, 14, 35, 10, 19, 31, 42











**struct Node\* insert(int arr[], struct Node\* root, int i, int n)**

**{**

**// Base case for recursion**

**if (i < n)**

**{**

**struct Node\* temp = newNode(arr[i]);**

**root = temp;**

**// insert left child**

**root->left = insert(arr, root->left, 2 \* i + 1, n);**

**// insert right child**

**root->right = insert(arr, root->right, 2 \* i + 2, n);**

**}**

**return root;**

**}**

| **insert(arr, root, 0, 7)** |
| --- |



**.**

| **insert(arr, root, 1, 7)** |
| --- |



| **insert(arr, root, 4, 7)** |
| --- |



| **insert(arr, root, 3, 7)** |
| --- |







| **insert(arr, root, 7, 7)** |
| --- |
| **insert(arr, root, 8, 7)** |





**TRAVERSALS**











INORDER----LVR

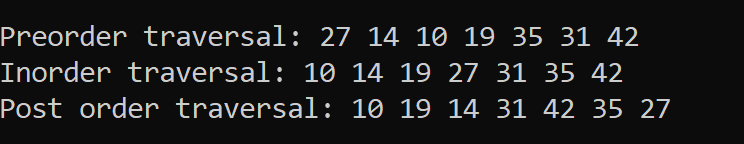
10 14 19 27 31 35 42

PREORDER----VLR

27 14 10 19 35 31 42

POSTORDER----VLR

10 19 14 31 42 35 27



**BINARY SEARCH TREE**

27, 14, 35, 10, 19, 31, 42



27, 14, 35, 10, 19, 31, 42



27, 14, 35, 10, 19, 31, 42





27, 14, 35, 10, 19, 31, 42









27, 14, 35, 10, 19, 31, 42











27, 14, 35, 10, 19, 31, 42











27, 14, 35, 10, 19, 31, 42











11, 6, 8, 19, 4 , 10, 5, 17, 43, 49,31















