Q)

Code:

#include <stdio.h>

void bubblesort(int arr[100],int size)

{

for(int i=0;i<size-1;i++)

{

for(int j=0;j<size-1-i;j++)

{

if(arr[j]>arr[j+1])

{

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

for(int i=0;i<size;i++)

{

printf("%d ",arr[i]);

}

}

int main()

{

int size, arr[100];

printf("\nEnter the array limit/size: ");

scanf("%d",&size);

printf("\nEnter the array elements: ");

for(int i=0;i<size;i++)

{

scanf("%d",&arr[i]);

}

printf("The sorted array is: ");

bubblesort(arr,size);

return 0;

}

Output:

Enter the array limit/size: 10

Enter the array elements: 23 75 81 34 92 34 34 10 90 98

The sorted array is: 10 23 34 34 34 75 81 90 92 98

**...Program finished with exit code 0**

**Press ENTER to exit console.**

#include <stdio.h>

struct BTNode

{

int data;

BTNode \*lc;

BTNode \*rc;

};

sturct BTNode\* insert(struct BTNode \*root)

{

struct BTNode \*newnode=(struct BTNode\*)malloc(sizeof)

printf("\nEnter the node data: ");

scanf("%d",&node->data);

}

void Inorder\_Traversal(struct BTNode \*root)

{

}

void Preorder\_Traversal(struct BTNode \*root)

{

}

void Postorder\_Traversal(struct BTNode \*root)

{

}

int Count\_Nodes(struct BTNode \*root)

{

}

int Count\_Leaf\_Nodes(struct BTNode \*root)

{

}

int main()

{

struct BTNode \*root=NULL;

printf("\nEnter the elements of the binary tree.");

root=insert(root);

printf("\nInorder traversal of the binary tree is: ");

Inorder\_Traversal(root);

printf("\nPreorder traversal of the binary tree is: ");

Preorder\_Traversal(root);

printf("\nPostorder traversal of the binary tree is: ");

Postorder\_Traversal(root);

int nodes=Count\_Nodes(root);

printf("\nNo of nodes in the binary tree is: ");

int leafnode=Count\_Leaf\_Nodes(root);

printf("\nNo of leaf nodes in the binary tree is: ");

return 0;

}

#include <stdio.h>

#include <stdlib.h>

struct BTNode

{

int data;

struct BTNode \*lc;

struct BTNode \*rc;

};

struct BTNode\* insert(struct BTNode \*root)

{

int data;

struct BTNode \*node=(struct BTNode\*)malloc(sizeof(struct BTNode));

printf("\nEnter the node data: ");

scanf("%d",&data);

if(data==-999999)

{

return NULL;

}

node->data=data;

root=node;

root->lc=insert(root);

root->rc=insert(root);

return root;

}

void Inorder\_Traversal(struct BTNode \*root)

{

if(root==NULL)

{

return;

}

Inorder\_Traversal(root->lc);

printf("%d ",root->data);

Inorder\_Traversal(root->rc);

return;

}

void Preorder\_Traversal(struct BTNode \*root)

{

if(root==NULL)

{

return;

}

printf("%d ",root->data);

Preorder\_Traversal(root->lc);

Preorder\_Traversal(root->rc);

return;

}

void Postorder\_Traversal(struct BTNode \*root)

{

if(root==NULL)

{

return;

}

Postorder\_Traversal(root->lc);

Postorder\_Traversal(root->rc);

printf("%d ",root->data);

return;

}

int Count\_Nodes(struct BTNode \*root)

{

if(root==NULL)

{

return 0;

}

if(root->lc==NULL && root->rc==NULL)

{

return 1;

}

return Count\_Nodes(root->lc)+Count\_Nodes(root->rc)+1;

}

int Count\_Leaf\_Nodes(struct BTNode \*root)

{

if(root==NULL)

{

return 0;

}

if(root->lc==NULL && root->rc==NULL)

{

return 1;

}

else

{

int ans1=Count\_Nodes(root->lc);

int ans2=Count\_Nodes(root->rc);

return ans1+ans2;

}

}

int main()

{

struct BTNode \*root=NULL;

printf("\nEnter the elements of the binary tree.");

root=insert(root);

printf("\nInorder traversal of the binary tree is: ");

Inorder\_Traversal(root);

printf("\nPreorder traversal of the binary tree is: ");

Preorder\_Traversal(root);

printf("\nPostorder traversal of the binary tree is: ");

Postorder\_Traversal(root);

int nodes=Count\_Nodes(root);

printf("\nNo of nodes in the binary tree is: %d",nodes);

int leafnode=Count\_Leaf\_Nodes(root);

printf("\nNo of leaf nodes in the binary tree is: %d",leafnode);

return 0;

}