***PROGRAM:***

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

struct node

{

int data;

struct node\*left;

struct node\*right;

};

typedef struct node s1;

s1 \*createBST(s1 \*,int);

void inorder(s1 \*);

void preorder(s1 \*);

void postorder(s1 \*);

s1 \*root=NULL;

void main()

{

int choice,item,i,n;

do

{

printf("\n1.create a binary search tree");

printf("\n2.inorder traversal\n3.preorder traversal\n4.post ordertraversal\n5.exit");

printf("\nenter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("\nenter the number of elements:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("\nenter the element:");

scanf("%d",&item);

root=createBST(root,item);

}

break;

case 2:

printf("\ninorder traversal");

inorder(root);

break;

case 3:

printf("\npreorder traversal");

preorder(root);

break;

case 4:

printf("\npostorder traversal");

postorder(root);

break;

case 5:

exit(0);

}

}

while(choice!=5);

}

s1 \*createBST(s1 \*root,int item)

{

if(root==NULL)

{

root=(s1 \*)malloc(sizeof(s1));

root->left=root->right=NULL;

root->data=item;

return root;

}

else

{

if(item<root->data)

root->left=createBST(root->left,item);

else if(item>root->data)

root->right=createBST(root->right,item);

else

printf("\nduplicate element");

}

return root;

}

void inorder(s1 \*root)

{

if(root!=NULL)

{

inorder(root->left);

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

inorder(root->right);

}

}

void preorder(s1 \*root)

{

if(root!=NULL)

{

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

preorder(root->left);

preorder(root->right);

}

}

void postorder(s1 \*root)

{

if(root!=NULL)

{

postorder(root->left);

postorder(root->right);

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

}

}

***OUTPUT:***

******

