

1. Write a program:
- To construct Binary Search tree
 - Traverse the tree using inorder, postorder, preorder
 - Display the elements in the tree.
- Program

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct BST {
```

```
    int data;
```

```
    struct BST *left, *right;
```

```
};
```

```
struct BST *root = NULL, *temp;
```

```
struct BST *create()
```

```
void create() { struct BST *create()
```

```
{ temp = (struct BST *) malloc (sizeof(struct BST));
```

```
    printf("Enter data: ");
```

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

```
    temp->left = temp->right = NULL;
```

```
    return temp;
```

```
} if (root == NULL)
```

```
{ root = temp; }
```

```
else
```

```
{ if (
```

```
void insert(struct BST *root, struct BST *temp)
```

```
{
```

```
    if (temp->data < root->data)
```

```
    { if (root->left != NULL)
```

```
        insert (root->left, temp);
```

```
    else root->left = temp;
```

```
}
```

```
if (temp->data > root->data)
{
    if (root->right != NULL)
        insert(root->right, temp);
    else
        root->right = temp;
}
}
```

```
void inorder(struct BST *root)
{
    if (root != NULL)
    {
        inorder(root->left);
        printf("%d\t", root->data);
        inorder(root->right);
    }
}
```

```
void postorder(struct BST *root)
{
    if (root != NULL)
    {
        postorder(root->left);
        postorder(root->right);
        printf("%d\t", root->data);
    }
}
```

```
void preorder(struct BST *root)
{
    if (root != NULL)
    {
        printf("%d\t", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
```


void main()

{ int choice; char ch;

printf("Enter operation\n1.create\n2.display using
inorder\n3.display using postorder\n4.display
using preorder\n5. -1 to end\n");

while(1)

{ printf("Enter operation:");
scanf("%d", &choice);

if (choice == -1)
{ return ; }

else

{ switch (choice)

{

case 1 : do

{ temp = create();

if (root == NULL)

root = temp;

else

insert(root, temp);

printf("Do you want to enter
more (Y/N)? ");

getchar();

scanf("%c", &ch);

} while (ch == 'y' || ch == 'Y');

case 2 : printf("elements of tree are:");

pr inorder (root);

case 3 : printf("elements of tree are:");

postorder (root);

case 4 : printf("elements of tree are:");

preorder (root);

```

        default: printf("invalid operation");
    }
}
}

```

Output:-

1. Insert data into BST
2. display the elements

Enter operation: 1

Enter data: 7

Enter operation: 1

Enter data: 9

Enter operation: 1

Enter data: 4

Enter operation: 1

Enter data: 6

Enter operation: 1

Enter data: 23

Enter operation: 2

Elements using Inorder Traversal: 4 6 7 9 23

Elements using Preorder Traversal: 7 4 6 9 23

Elements using Postorder Traversal: 6 4 23 9 7

C:\Users\hp\OneDrive\Desktop\22cs300\BST.exe

1.Enter data into BST

2.To stop

Enter choice: 1

Enter data: 7

Enter choice: 1

Enter data: 9

Enter choice: 1

Enter data: 4

Enter choice: 1

Enter data: 6

Enter choice: 1

Enter data: 23

Enter choice: 2

In-order Traversal: 4 6 7 9 23

Pre-order Traversal: 7 4 6 9 23

Post-order Traversal: 6 4 23 9 7

Process exited after 33.94 seconds with return value 0

Press any key to continue . . .