



WAP

- a) To construct a binary search tree
- b) To traverse the tree using all the methods
 - 1) in-order
 - 2) pre-order
 - 3) post-order
- c) To display elements of the tree.

Code :-

```
#include <stdio.h>
#include <stdlib.h>
struct bnode
{
    int value;
    struct bnode *l;
    struct bnode *r;
} *root = NULL, *temp = NULL, *t2, *t1;

void insert();
void inorder (struct bnode *t);
void create();
void search (struct bnode *t);
void preorder (struct bnode *t);
void postorder (struct bnode *t);

int flag = 1;
```

```
void main ()
```

```
{
```

```
int ch;
```

```
while (1) {
```

```
    printf(" 1 - insert \n");
```

```
    printf(" 2 - inorder \n");
```

```
    printf(" 3 - preorder \n");
```

```
    printf(" 4 - postorder \n");
```

```
    printf(" 5 - Exit \n");
```

```
    printf(" Enter your choice \n");
```

```
    scanf("%d", &ch);
```

```
    switch (ch);
```

```
{
```

```
    case 1:
```

```
        insert();
```

```
        break;
```

```
    case 2:
```

```
        inorder (root);
```

```
        break;
```

```
    case 3:
```

```
        preorder (root);
```

```
        break;
```

```
    case 4:
```

```
        postorder (root);
```

```
        break;
```

```
    case 5: exit(0);
```



default :

```
printf("Invalid choice\n");
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
void create() {
```

```
int data;
```

```
printf("Enter data of node to be inserted:");
```

```
scanf("%d", &data);
```

```
temp = (struct bnode*) malloc( sizeof( struct bnode));
```

```
temp->value = data;
```

```
temp->l = temp->r = NULL;
```

```
}
```

```
void insert()
```

```
{
```

```
create();
```

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

```
root = temp;
```

```
else
```

```
search (root);
```

```
}
```

```
void search ( struct bnode *t)
```

```
{
```

```
if ((temp->value > t->value) && (t->r != NULL))
```

```
search (t->r);
```


else if (Ctemp → value > t → value) && (t → r == NULL))

t → r = temp;

else if (Ctemp → value < t → value) && (t → l == NULL))

search(t → l);

else if (Ctemp → value < t → value) && (t → l == NULL))

t → l = temp;

}

void inorder (struct bnode *t)

{

if (root == NULL)

{

printf("No elements in the tree\n");

return;

}

printf("%d", t → value);

if (t → l == NULL)

inorder (t → r);

}

void preorder (struct bnode *t)

{

if (root == NULL) {

printf("No elements in tree\n");

return;

}



```
printf(" %d → ", t → value);  
if (t → l == NULL)  
    preorder (t → l);  
if (t → r != NULL)  
    preorder (t → r);  
}
```

```
void postorder (struct bnode *t)  
{  
    if (root == NULL) {  
        printf(" No elements in the tree\n");  
        return;  
    }  
    if (t → l == NULL)  
        postorder (t → l);  
    if (t → r != NULL)  
        postorder (t → r);  
    printf(" %d → ", t → value);  
}
```

Expected output :-

1. Insert

2. Inorder

3. Preorder

4. Postorder

5. Exit

Enter your choice : 1

Enter data to be inserted : 15

Enter your choice : 1

Enter data to be inserted : 2

Enter your choice : 1

Enter data to be inserted : 67

Enter your choice : 1

Enter data to be inserted : 1

Enter your choice : 1

Enter data to be inserted : 90

Enter your choice : 1

Enter data to be inserted : 34

Enter your choice: 2

1 → 2 → 15 → 34 → 67 → 90 →

Enter your choice: 3

15 → 2 → 1 → 67 → 34 → 90 →

Enter your choice: 4

1 → 2 → 34 → 90 → 67 → 15 →