15/2/24

LAB - 8

WAP (i) To construct a binary Search tree.
(ii) To traverse tree using all the methods i.e in-order, preorder and postorder
(iii) To display the elements in the tree

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

struct node
{
      int data;
      struct node *left;
      struct node *right;
};
struct node *root=NULL;
void create()
{
      int val;
      struct node *new-node = (struct node*)
                  malloc (sizeof(struct node));
      struct node *ptr = root;
      struct node *ptr1 = NULL;
      printf("Enter data:");
      scanf("%d", &val);
      new-node -> left = NULL;
      new-node -> right = NULL;
      new-node -> data = val;
      if (root == NULL)
      {     root = new-node;
      }
      else
      {
          while(ptr != NULL)
          {       ptr1 = ptr;
                  if(ptr -> data > val)
                        ptr = ptr -> left;
                  else
                        ptr = ptr -> right;
          }
          if (ptr1 -> data > val)
                  ptr1 -> left = new_node;
          else
                  ptr1 -> right = new_node;
      }
}

void inorder(struct node *ptr)
{
    if (ptr! = NULL)
    {
        inorder (ptr -> left);
        printf(" %d ", ptr -> data);
        inorder (ptr -> right);
    }
}

void pre_order (struct node *ptr)
{
    if(ptr != NULL)
    {   printf("%d ", ptr -> data);
        pre_order (ptr -> left);
        pre_order (ptr -> right);
    }
}
```

N
15/2/24

```
void post-order (struct node *ptr)
{
        if (ptr != NULL)
        {
                post-order (ptr -> left);
                post-order (ptr -> right);
                printf ("%d  ", ptr -> data);
        }
}

void main()
{
        int n;
        printf ("Enter no. of nodes: ");
        scanf ("%d ", &n);
        for (int i=0 ; i<n ; i++)
        {       create();
        }
        printf ("\n Inorder: \n");
        inorder (root);
        printf ("\n PreOrder : \n");
        pre-order (root);
        printf (" \n PostOrder : \n");
        post-order (root);
}
```

OUTPUT

```
Enter no. of nodes: 6
Enter data: 5
Enter data: ~~9~~ 3
Enter data: 4
Enter data: 9
Enter data: 6
Enter data: 8
Inorder:
3  4   5    6  8   9
Preorder:
5  3   4   9   6   8
Postorder:
4  3   8   6   9  5
```