Lab10.c

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

 struct node
 {
   int key;
   struct node *left;
   struct node *right;
   };

 struct node *root;
  {
     struct node *temp;
     temp = (struct node* ) malloc (size of (struct node));
       temp -> key = data;
       temp -> left = temp -> right= NULL;
        return temp;
         }

   void  insert(struct node *root, struct node *temp)
       {
         if ( temp -> key <root -> key)
                {
                 ~~if (temp -> key < root -> key)~~
                   {
                   if (root -> left != NULL)
                     {
                        insert (root -> left, temp);
                     }
```

Scanned with CamScanner

```
else
{            ~~right~~
   root -> ~~left~~ = temp;
       }     left
}

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

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

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

      void preorder (struct node * root )
        {
           if (root ! = NULL)
              {
                 postorder (root -> left);
                 postorder (root -> right);
                 ~~post~~ printf("%d \t", root -> key);
                   }
              }
            int main ()
              {
                 char ch;
                 struct node *temp;
                   root = NULL;
                   int Choice = 0;
                   int data ;

        while (1)
          {
             printf (" \n\n ******MENU****** \n");
              printf(" Choose an option from the list: ");
              printf(" \n [1] CREATE A TREE \n
                                 [2] INORDER TRAVERSAL \n
                                 [3] PREORDER TRAVESAL\n
                                 [4] POSTORDER TRAVORSAL\n
                                 [5] DISPLAY \n
                                 [6] EXIT\n");
```

Scanned with CamScanner

```
printf("\nEnter your choice:");
 scanf("%d", &choice);
 switch(choice)
 {
   case 1: do {
          printf("\nEnter the value:");
           scanf("%d",&data);
          if(root==NULL)
            {
              root = temp;
             }
          else
          {
            insert(root,temp);
           }
        printf("\nDo you want to Enter
                   more (Y/N)? ");
        getchar();
         scanf("%c",&ch);
        } while(ch=='y'||ch=='Y');
          break;

case 2: printf("\n Inorder Traversal --->\t");
          inorder(root);
          break ;

case 3: printf("\n PREORDER TRANSVERSAL-
          preorder(root);
          break;
```

```
case 4: printf("
          p

 case 5: display

  case 6: EXIT(
   default ;
     printf(
    }
    }
   return 0;
  }
```

Scanned with CamScanner

```
case 4: printf(" \n POSTORDER TRANVERSAL -->\t");
                postorder(root);
                  break;

 case 5: display(root);
              break;

  case 6: EXIT(1);
    default ;
       printf("\n INVALID. CHOICE ");
     }
     }
    return 0;
   }
```

Scanned with CamScanner