AVIRATH HEGDE
1BM19CS195

21-12-2020

LAB-10 : Binary Search Tree Implementation.

A.
```
Struct node
{
    int key;
    Struct node * left;
    Struct node * right;
}

Struct node* create(int data)
{
    Struct node *temp;
    temp = (Struct node *) malloc (sizeof (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 (root -> left != NULL)
            insert (root -> left, temp);
        else
            root -> left = temp;
    }
```

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

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

Void Pmonorder (struct node *root)
{
  if (root != NULL)
{
  Printf ("%d", root -> key);
  Pmonorder (root -> left);
  Pmorder (root -> right);
    }
}
```

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

int main()
{
  char ch;
  struct node *root = NULL, *temp;
  do{
   temp = create(data);
   if(root == NULL)
      root = temp;
  else
     insert(root, temp);
  printf("Do you want to enter name (Y/N)?");
   getchar();
   scanf("%c", &ch);
} while(ch == 'y' || ch == 'Y');
  return 0;
}
```