M.SIRI ANJANI KEERTHI

AP19110010439

CSE-(F)

DSA LAB PROGRAMS

1.Write a C program to print preorder, inorder, and postorder traversal on Binary Tree.

```
#include <stdio.h>

#include <stdlib.h>

void Postorder();

void Inorder();

void Preorder();

struct node

{
```

```
        int data;
        struct node* left;
        struct node* right;


};  struct node* newNode(int data)
{
        struct node* node = (struct node*)
        malloc(sizeof(struct node));
        node->data = data;
        node->left = NULL;
        node->right = NULL;

        return(node);
}  void Postorder(struct node* node) {
        if (node == NULL)
        return;
```

```
        Postorder(node->left);
        Postorder(node->right);
        printf("%d ", node->data);
} void Inorder(struct node* node) {
        if (node == NULL)
        return;
        Inorder(node->left);


        printf("%d ", node->data);
        Inorder(node->right);
} void Preorder(struct node* node) {
        if (node == NULL)
        return;
        printf("%d ", node->data);
        Preorder(node->left);
        Preorder(node->right);
```

```
} void main()
{
        struct node *root = newNode(23);
        root->left = newNode(6);
        root->left->left = newNode(20);
        root->left->right = newNode(17);
        root->right = newNode(3);
        root->right->left = newNode(5);
        root->right->right = newNode(8);
        printf("\nPreorder traversal of binary tree is \n");
        printPreorder(root);
        printf("\nInorder traversal of binary tree is \n");

        printInorder(root);
        printf("\nPostorder traversal of binary tree is \n");
        printPostorder(root);
```

}

OUTPUT:

Preorder traversal of
binary tree is

23 6 20 17 3 5 8

Inorder traversal of
binary tree is

20 6 17 23 5 3 8

Postorder traversal of
binary tree is

20 17 6 5 8 3 23

2.Write a C program to create (or insert) and inorder traversal on Binary Search Tree.

```
#include<stdio.h>

#include<stdlib.h>

typedef struct node

{
```

```
  int data;
  struct node *left;
  struct node *right;
} node;
node *create()
{
        node *s;
        int x;
        printf("Enter data(-1 for no node):");


        scanf("%d",&x);
        if(x==-1)
 return NULL;
        s=(node*)malloc(sizeof(node));
        s->data=x;
        printf("Enter left child of %d:\n",x);
        s->left=create();
```

```
        printf("Enter right child of %d:\n",x);
        s->right=create();
        return s;
} void inorder(node *t)
{
  if(t!=NULL)
  {
        inorder(t->left);
        printf(" %d",t->data);
        inorder(t->right);
  }
} void main()

{
  node *root;
  root=create();
```

```
  printf("\nThe inorder traversal of tree is: ");
  inorder(root);
}
```

OUTPUT:

```
Enter data(-1 for no node):23
Enter left child of 23:
Enter data(-1 for no node):6
Enter left child of 6:
Enter data(-1 for no node):15
Enter left child of 15:
Enter data(-1 for no node):-1
Enter right child of 89:
Enter data(-1 for no node):-1
Enter right child of 38:
Enter data(-1 for no node):-1
Enter left child of 67:
Enter data(-1 for no node)
```

3.Write a C program for linear search algorithm.

```
#include <stdio.h>


void main()

{

  int array[75], search, x, n;

  printf("Enter number of elements in array\n");

  scanf("%d", &n);

  printf("Enter %d integer(s)\n", n);

  for (x = 0; x< n; x++)

{
        scanf("%d", &array[x]);

  printf("Enter a number to search\n");

  scanf("%d", &search);}
```

```
  for (x = 0; x < n; x++)
  {
        if (array[x] == search)
        { printf("%d is present at location %d.\n", search, x+1);
        break;
        }
  }
if (x == n){
        printf("%d isn't present in the array.\n", search);}
        }
```

OUTPUT:

Enter number of elements in array

5

Enter 3 integer(s)

14

49

38

56

75

Enter a number to search

56

56 is present at location 4.

4.Write a C program for binary search algorithm

```
#include <stdio.h>


int main()



{
  int i, start, end, middle, n, search, array[100];
  printf("Enter number of elements\n");
  scanf("%d", &n);
```

```
printf("Enter %d integers\n", n);

for (i = 0; i < n; i++)

        scanf("%d", &array[i]);

printf("Enter value to find\n");

scanf("%d", &search);

start = 0;

end = n - 1;

middle = (start+end)/2;

while (start <= end) {

        if (array[middle] < search)

        start = middle + 1;

        else if (array[middle] == search) {

        printf("%d found at location %d.\n", search, middle+1);

        break;

        } else

        end = middle - 1;

        middle = (start + end)/2;
```

```
  } if (start > end)

        printf("Not present: %d isn't present in the list.\n", search);

  return 0;

}
```

OU
TP
UT
:

En
ter
nu
mb
er
of
ele
me
nts

4

En
ter
3
int
eg

ers

17

20

23

6

Enter value to find

23

23 found at location 3.