

# LAB WORK

SRAVIKA-AP19110010423

1) Write a Program to print preorder, inorder, and postorder traversal on Binary Tree.

**Code:**

```
#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(1);
    root->left      = newNode(2);
    root->right     = newNode(3);
    root->left->left = newNode(4);
    root->left->right = newNode(5);

    printf("\nPreorder traversal of binary tree is \n");
    Preorder(root);
    printf("\nInorder traversal of binary tree is \n");
    Inorder(root);
    printf("\nPostorder traversal of binary tree is \n");
    Postorder(root);

}

```

### **Output:**

Preorder traversal of binary tree is

1 2 4 5 3

Inorder traversal of binary tree is

4 2 5 1 3

Postorder traversal of binary tree is

4 5 2 3 1

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

### **Code:**

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

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

```
typedef struct node
```

```
{
```

```
    int data;
```

```
struct node *left;
struct node *right;
} node;
```

```
node *create()
{
    node *p;
    int x;
    printf("Enter data(-1 for no node):");
    scanf("%d",&x);

    if(x==-1)
        return NULL;

    p=(node*)malloc(sizeof(node));
    p->data=x;
    printf("Enter left child of %d:\n",x);
    p->left=create();
    printf("Enter right child of %d:\n",x);
    p->right=create();
    return p;
}
```

```
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):45
Enter left child of 45:
Enter data(-1 for no node):68
Enter left child of 68:
Enter data(-1 for no node):90
Enter left child of 90:
Enter data(-1 for no node):-1
Enter right child of :90
Enter data(-1 for no node):-1
Enter right child of 68:
Enter data(-1 for no node):-1
Enter left child of 45:

```

3)Write a C program for linear search algorithm.

**Code:**

```

#include <stdio.h>
void main()
{ int num;

    int i,search, flag = 0;

    printf("Enter the number of elements ");
    scanf("%d", &num);
    int array[num];
    printf("Enter the elements \n");
    for (i = 0; i < num; i++)
    {
        scanf("%d", &array[i]);
    }

    printf("Enter the element to be searched ");
    scanf("%d", &search);

    for (i = 0; i < num ; i++)
    {

```

```

        if (search == array[i] )
        {
            flag = 1;
            break;
        }
    }
    if (flag == 1)
        printf("Element is present in the array at position %d",i+1);
    else
        printf("Element is not present in the array\n");
}

```

**Output:**Enter the number of elements 6.

Enter the elements

4

5

8

7

3

2

Enter the element to be searched 8

Element is present in the array at position 3

4)Write a C program for binary search algorithm.

**Code:**

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

```
int main()
```

```
{
```

```
    int arr[50],i,n,x,flag=0,first,last,mid;
```

```
    printf("Enter size of array:");
```

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

```
    printf("\nEnter array element(ascending order)\n");
```

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

```
    {
```

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

```
    }
```

```

printf("\nEnter the element to search:");
scanf("%d",&x);

first=0;
last=n-1;

while(first<=last)
{
    mid=(first+last)/2;

    if(x==arr[mid])
    {
        flag=1;
        break;
    }
    else
        if(x>arr[mid])
            first=mid+1;
        else
            last=mid-1;
}

if(flag==1)
    printf("\nElement found at position %d",mid+1);
else
    printf("\nElement is not found");

return 0;
}

```

**Output:**

Enter size of array:

6

Enter array elements (ascending order )

40

76

82

97

100

112

Enter the element to search:82

Element found at position 3