**Practical-9**

**Implementation of Tree and Searching**

1. Write algorithm and implement program for creating a binary tree, traversing a binary tree using preorder, inorder and postorder.

* Code:

#include <stdio.h>

#include <stdlib.h>

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 printPostorder(struct node\* node)

{

if (node == NULL)

{

return;

}

printPostorder(node->left);

printPostorder(node->right);

printf("%d ", node->data);

}

void printInorder(struct node\* node)

{

if (node == NULL)

{

return;

}

printInorder(node->left);

printf("%d ", node->data);

printInorder(node->right);

}

void printPreorder(struct node\* node)

{

if (node == NULL)

{

return;

}

printf("%d ", node->data);

printPreorder(node->left);

printPreorder(node->right);

}

int main()

{

int rt,l,r,ll,lr;

printf("Enter Root Node");

scanf("%d",&rt);

struct node\* root = newNode(rt);

printf("Enter Left Of Root Node");

scanf("%d",&l);

root->left = newNode(l);

printf("Enter Right Of Root Node");

scanf("%d",&r);

root->right = newNode(r);

printf("Enter Left Left Of Root Node");

scanf("%d",&ll);

root->left->left = newNode(ll);

printf("Enter Left Right Of Root Node");

scanf("%d",&lr);

root->left->right = newNode(lr);

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);

printf("\n");

getchar();

return 0;

}

* Output:



1. Write a program for searching an element in an array using linear search.

* Code:

#include <stdio.h>

int main()

{

int array[50], search, c, n;

printf("Enter How Many Number You Want To Sort : \n");

scanf("%d", &n);

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

{

printf("\nEnter %d Number:",c+1);

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

}

printf("Enter The Number That You Want to Search: \n");

scanf("%d", &search);

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

{

if (array[c] == search)

{

printf("%d is Found at location %d.\n", search, c+1);

break;

}

}

if (c == n)

printf("Not found! %d is not available in array.\n", search);

return 0;

}

* Output:



1. Write a program to implement binary search.

* Code:

#include<stdio.h>

int main()

{

int c, first, last, middle, n, search, array[100];

printf("Enter How Many Number You Want To Short : \n");

scanf("%d",&n);

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

{

printf("\nEnter %d Number:",c+1);

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

}

printf("enter The Number that you Want to SEARCH : \n");

scanf("%d", &search);

first = 0;

last = n - 1;

middle = (first+last)/2;

while (first <= last)

{

if (array[middle] < search)

{

first = middle + 1;

}

else if (array[middle] == search)

{

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

break;

}

else

last = middle - 1;

middle = (first + last)/2;

}

if (first > last)

printf("Not found! %d is not available in array.\n", search);

return 0;

}

* Output:

