V.Sai Deepika

AP19110010415

CSE-H.

1.Write a c program to print preorder ,inorder and post order traversal on binary tree

#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()

{

     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");

     printPreorder(root);

     printf("\nInorder traversal of binary tree is \n");

     printInorder(root);

     printf("\nPostorder traversal of binary tree is \n");

     printPostorder(root);

     getchar();

     return 0;

}

Output 1:

Preorder of binary tree is:

1 2 4 5 3

In order of binary tree is:

4 2 5 1 3

Post order of binary tree is:

4 5 2 3 1

2.Write a c program tp create (or insert) & inorder traversal on binary tree ?

#include<stdio.h>

#include<stdlib.h>

struct node

{

int key;

struct node \*left, \*right;

};

struct node \*newNode(int item)

{

struct node \*temp = (struct node \*)malloc(sizeof(struct node));

temp->key = item;

temp->left = temp->right = NULL;

return temp;

}

void inorder(struct node \*root)

{

if (root != NULL)

{

inorder(root->left);

printf("%d \n", root->key);

inorder(root->right);

}

}

struct node\* insert(struct node\* node, int key)

{

if (node == NULL) return newNode(key);

if (key < node->key)

node->left = insert(node->left, key);

else if (key > node->key)

node->right = insert(node->right, key);

return node;

}

int main()

{

struct node \*root = NULL;

root = insert(root, 25);

insert(root, 48);

insert(root, 14);

insert(root, 65);

insert(root, 28);

insert(root, 36);

insert(root, 42);

inorder(root);

return 0;

}

Output2

14

25

28

36

42

48

65

3.write a c program to linear search algorithm

#include<stdio.h>

int linear\_search(int arr[],int ele,int n);

void main()

{

int n,ele;

printf(“enter the size of array:”);

scanf(“%d”,&n);

int arr[n];

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

{

printf(“enter any integer:”);

scanf(“%d”,&arrr[i]);

}

printf(“enter the element to search:”);

scanf(“%d”,&ele);

linear\_search(arr,ele,n);

}

int linear \_search(int arr[],int ele,int n);

{

int found=0;

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

{

If(arr[i]==ele)

found=1;

}

If(found==1)

printf(“element is found”);

else

printf(“element is not found”);

}

Output:

Ennter size of elements in array

5

Enter any integer:

25

14

36

95

38

Enter the element to search:

95

Element is found.

4.write a c program to binary search algorithm?

#include<stdio.h>

int main()

{

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

printf(“enter number of elements\n”);

scanf(“%d”,&n);

printf(“enter %d integers\n”,n);

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

scanf(“%d”,&array[c]);

printf(“enter value to find\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 found at location %d.\n”,seach,middle+1);

break;

}

else

last=middle-1;

middle=(first+last)/2;

}

If (first>last)

printf(“not found %d isn’t present in the list.\n”,search);

return 0;

Output

Enter number of elements

5

Enter 5 integers

25

36

41

51

95

51 found at location 4.