Program

```c
#include<stdio.h>
int a[100];
void buildtree(int a[],int i,int item){
    int ch;
    int value;
    a[i]=item;
    printf("Do you want to create left child for %d  (1/0) :",item);
    scanf("%d",&ch);
    if(ch==1){
        printf("Enter the left child :");
        scanf("%d",&value);
        buildtree(a,2*i,value);
    }
    printf("Do you want to create right child for %d  (1/0) :",item);
    scanf("%d",&ch);
    if(ch==1){
        printf("Enter the right child :");
        scanf("%d",&value);
        buildtree(a,2*i+1,value);
    }
}
void preorder(int a[],int index ){
    if(index>=100||a[index]==-1)
        return;
    else{
        printf("%d\t",a[index]);
        preorder(a,2*index);
        preorder(a,2*index+1);
    }
}
void inorder(int a[],int index ){
    if(index>=100||a[index]==-1){
        return;
    }
    else{
        inorder(a,2*index);
        printf("%d\t",a[index]);
        inorder(a,2*index+1);
    }
}
void postorder(int a[],int index ){
    if(index>=100||a[index]==-1){
        return;
    }
    else{
        postorder(a,2*index);
        postorder(a,2*index+1);
        printf("%d\t",a[index]);
    }
}
void creation(){
    int i;
    int root;
    for(i=0;i<100;i++){
        a[i]=-1;
    }
    printf("Enter the root node :");
    scanf("%d",&root);
    buildtree(a,1,root);
    printf("TREE IS :");
```

Experiment No: 26
Date:

BINARY TREE USING ARRAY

Aim:

To implement a binary tree using array and perform preorder, inorder and postorder tree traversal .

Algorithm:

1. Start
2. Declare an array 'a' of size 100
3. Define function creation().

```
int i;
int root;
for(i=0;i<100;i++)
    a[i]=-1;
printf("Enter the root node :");
scanf("%d",&root);
buildtree(a,1,root);
printf("TREE IS :");
for(i=1;i<=20;i++)
    if(a[i]==-1)
        printf("- ");
    else
        printf("%d ",a[i]);
    end if
end for
```

4. Define function buildtree(int a[],int i,int item).

```
int ch;
int value;
a[i]=item;
printf("Do you want to create left child for %d  (1/0) :",item);
scanf("%d",&ch);
if(ch==1)
    printf("Enter the left child :");
    scanf("%d",&value);
    buildtree(a,2*i,value);
end if
printf("Do you want to create right child for %d  (1/0) :",item);
scanf("%d",&ch);
```

```
    for(i=1;i<=20;i++){
        if(a[i]==-1)
            printf("- ");
        else
            printf("%d ",a[i]);
    }
}
void main(){
    int c;
    char ch;
    do{
        printf("The operations are \n");
        printf("1.Creation\n2.Preorder\n3.Inorder\n4.Postorder\n");
        printf("Enter your choice :");
        scanf("%d",&c);
        switch(c){
            case 1:
                creation();
                break;
            case 2:
                preorder(a,1);
                break;
            case 3:
                inorder(a,1);
                break;
            case 4:
                postorder(a,1);
                break;
            default:
                printf("Invalid entry");
        }
        printf("\nDo you want to continue (y/n) ? :");
        scanf(" %c",&ch);
    }
    while(ch=='y'||ch=='Y');
}
```

Output

```
The operations are
1.Creation
2.Preorder
3.Inorder
4.Postorder
Enter your choice :1
Enter the root node :1
Do you want to create left child for 1  (1/0) :1
Enter the left child :2
Do you want to create left child for 2  (1/0) :1
Enter the left child :4
Do you want to create left child for 4  (1/0) :0
Do you want to create right child for 4  (1/0) :0
Do you want to create right child for 2  (1/0) :1
Enter the right child :5
Do you want to create left child for 5  (1/0) :0
Do you want to create right child for 5  (1/0) :0
Do you want to create right child for 1  (1/0) :1
Enter the right child :3
Do you want to create left child for 3  (1/0) :1
Enter the left child :6
```

```
if(ch==1)
    printf("Enter the right child :");
    scanf("%d",&value);
    buildtree(a,2*i+1,value);
end if
```

5. Define function preorder(int a[],int index).

```
if(index>=100||a[index]==-1)
    return;
else
    printf("%d\t",a[index]);
    preorder(a,2*index);
    preorder(a,2*index+1);
end if
```

6. Define function inorder(int a[],int index).

```
if(index>=100||a[index]==-1)
    return;
else
    inorder(a,2*index);
    printf("%d\t",a[index]);
    inorder(a,2*index+1);
end if
```

7. Define function postorder(int a[],int index).

```
if(index>=100||a[index]==-1)
    return;
else
    postorder(a,2*index);
    postorder(a,2*index+1);
    printf("%d\t",a[index]);
end if
```

8. In main()

```
Declare integer variable c and character variable char
Begin the do while loop
    display the options
    accept the choice from the user as ch
    switch(ch)
        case 1:
            creation()
        case 2:
            preorder(a,1)
        case 3:
            inorder(a,1)
        case 4:
            postorder(a,1)
        default:
            print "Invalid entry"
    Take input from user to continue or not as char
close while loop if char!=(y/Y)
```

9. Stop.

```
Do you want to create left child for 6  (1/0) :0
Do you want to create right child for 6  (1/0) :0
Do you want to create right child for 3  (1/0) :0
TREE IS :1 2 3 4 5 6 - - - - - - - - - - - - - -
Do you want to continue (y/n) ? :y
The operations are
1.Creation
2.Preorder
3.Inorder
4.Postorder
Enter your choice :2
1 2 4 5 3 6
Do you want to continue (y/n) ? :y
The operations are
1.Creation
2.Preorder
3.Inorder
4.Postorder
Enter your choice :3
4 2 5 1 6 3
Do you want to continue (y/n) ? :y
The operations are
1.Creation
2.Preorder
3.Inorder
4.Postorder
Enter your choice :4
4 5 2 6 3 1
Do you want to continue (y/n) ? :n
```

Result:

Program has been executed successfully and obtained the output.