18 12 20

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

struct node
{
    int data;
    struct node* left, *right;
};

struct node *newnode (int ele)
{
    struct node *temp = (struct node *) malloc (
                        sizeof (struct node));
    temp -> data = ele;
    temp -> left = temp -> right = NULL;
    return temp;
}
void inorder (struct node *root)
{
    if (root != NULL)
    {
        inorder (root -> left);
        printf ("%d", root -> data);
        inorder (root -> right);
    }
}
void preorder (struct node *node)
{
```

```
printf ("%d", node -> data);
 preorder (node -> left);
 preorder (node -> right);
}
void postorder ( struct node * node)
{
   if (node == NULL)
  return;
  postorder (node -> left);
  postorder (node -> right);
   printf ("%d", node -> data);
}
 struct node* addtotree ( struct node * node, int data)
{
   if (node == NULL)
   return; newnode (data);
  if ( data < node -> data )
   node -> left = addtotree (node -> left, data);
   else if ( data > node -> data)
    node -> right = addtotree (node -> right, data);
    return node;
}

   int main( )
{
   printf (" Enter number of elements \n");
   int n, choice;
   scanf ("%d", &n);
   int arr [n]
   for [ int i=0 ; i<n ; i++)
{
```

```
    addtotree (root, arr[i]);
}
    do {
     printf ("\n 1. Preorder \n
                2. Postorder \n
                3. Inorder \n
                4. Exit \n" );
     printf ( " Enter choice : " );
     scanf ( "%d", & choice );
     switch ( choice )
{
     case 1 : preorder (root );
              break ;
     case 2 : postorder (root);
              break;
     case 3 :  inorder (root);
              break;
     case 4 ; return 0;
     }
}
   while ( choice ! = 4 )
     return 0;
}
```