```
#include<stdlib.h>
#include<stdio.h>
    struct node
   {
       int data;
       struct node *left, *right;
    };
    /* Given a binary tree, print its nodes in inorder*/
void printInorder(struct node* node)
{
   if (node == NULL)
      return;

   /* first recur on left child */
   printInorder(node->left);

   /* then print the data of node */
   printf("%d ", node->data);
 /* now recur on right child */
   printInorder(node->right);
}


struct node *create()
{
  struct node *temp, *root;
  int data, choice;
  temp = (struct node *)malloc(sizeof(struct node));
  printf("\n Press 0 to exit");
  printf("\nPress 1 for new node");
  printf("\n Enter your choice : ");
  scanf("%d", &choice);
  if(choice==0)
   {
    return 0;
   }
   else
   {
  printf("Enter the data:");
  scanf("%d", &data);
  temp->data = data;
  printf("\n Enter the left child of %d", data);
  temp->left = create();
  printf("\n Enter the right child of %d", data);
  temp->right = create();
   return temp;
}
}
void main()
   {
      struct node *root;
      printf("creating tree\n ");
      root = create();

      // Function call
   printf("\nInorder traversal of binary tree is \n");
   printInorder(root);
}
```