```
/* C program to print preorder, inorder, and postorder traversal on Binary tree */

#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node* left;
   struct node* right;
};

struct node* newNode(int value)
{
   struct node* node = (struct node*)malloc(sizeof(struct node));
   node->data = value;
   node->left = NULL;
   node->right = NULL;
   return node;
}

void Preorder(struct node* node)
{
   if (node == NULL)
   return;
   printf("%d ->", node->data);
   Preorder(node->left);
   Preorder(node->right);
}

void Inorder(struct node* node)
{
   if (node == NULL)
   return;
   Inorder(node->left);
   printf("%d ->", node->data);
   Inorder(node->right);
}

void Postorder(struct node* node)
{
   if (node == NULL)
   return;
```

```
    Postorder(node->left);
    Postorder(node->right);
    printf("%d ->", node->data);
}

int main()
{
    struct node *root      = newNode(9);
    root->left             = newNode(8);
    root->right            = newNode(7);
    root->left->left       = newNode(6);
    root->left->right      = newNode(5);

    printf("\nPreorder traversal of binary tree is \n");
    Preorder(root);

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

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

    getchar();
    return 0;
}
```

Output:

```
Preorder traversal of binary tree is
9 ->8 ->6 ->5 ->7 ->
Inorder traversal of binary tree is
6 ->8 ->5 ->9 ->7 ->
Postorder traversal of binary tree is
6 ->5 ->8 ->7 ->9 ->
```

```
/* C program to create (or insert) and inorder traversal on Binary search tree */

#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node* left;
```

```
    struct node* right;
};

struct node* createNode(int value)
{
    struct node* newNode = malloc(sizeof(struct node));
    newNode->data = value;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}


void Inorder(struct node* node)
{
    if (node == NULL)
    return;
    Inorder(node->left);
    printf("%d ->", node->data);
    Inorder(node->right);
}

void main()
{
    struct node* root     =createNode(9);
    root->left            =createNode(8);
    root->right           =createNode(7);
    root->left->left      =createNode(6);
    root->left->right     =createNode(5);

    printf("\nInorder traversal \n");
    Inorder(root);
}
```

Output:

```
Inorder traversal
6 ->8 ->5 ->9 ->7 ->
```

/* C program for linear search algorithm */

```
#include<stdio.h>
int main()
{
  int arr[10], se, i, n;
  printf("Enter the number of elements to be used in an array:\n");
  scanf("%d",&n);
  printf("Enter %d numbers\n", n);
  for ( i = 0 ; i < n ; i++ )
    scanf("%d",&arr[i]);
  printf("Enter the number to search\n");
  scanf("%d",&se);
  for ( i = 0 ; i < n ; i++ )
  {
    if ( arr[i] == se )
    {
      printf("%d is present at location %d.\n", se, i+1);
      break;
    }
  }
  if ( i == n )
    printf("%d is not present in array.\n", se);
  return 0;
 }
```

Output:

```
Enter the number of elements to be used in an array:
5
Enter 5 numbers
1
2
3
4
5
Enter the number to search
4
4 is present at location 4.
```

```
/* C program for binary search algorithm*/
```

```c
#include<stdio.h>
int main()
{
    int arr[10], se, i, n, found=0, top, mid, bot;
   printf("Enter the number of elements to be used in an array:\n");
   scanf("%d",&n);
   printf("Enter %d numbers\n", n);
   for ( i = 0 ; i < n ; i++ )
     scanf("%d",&arr[i]);
   printf("Enter the number to search\n");
   scanf("%d",&se);
   top=0;
   bot=n-1;

   while(top<=bot)
   {
      mid = (top+bot) / 2 ;
       if(arr[mid]==se)
       {
          found=1;
          break;
       }
       else if (arr[mid]>se)
       {
          bot=mid-1;
       }
       else if(arr[mid]<se)
       {
          top=mid+1;
       }
   }

   if(found==1)
   {
      printf("%d is present at location %d.\n", se, mid+1);
   }
   else
   {
      printf("%d is not present in array.\n", se);
   }
   return 0;
}
```

Output:

```
Enter the number of elements to be used in an array:
5
Enter 5 numbers
1
2
3
4
5
Enter the number to search
5
5 is present at location 5.
```