Question-1

```
#include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node* right;

struct node* left;

};

struct node* newNode(int data)

{

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

node->data = data;


node->right = NULL;

node->left = NULL;

return(node);

}


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);

}

void preorder(struct node* node)

{

if (node == NULL)

return;

printf("%d ", node->data);

preorder(node->left);

preorder(node->right);

}

int main()

{

struct node *root = newNode(2);

root->left = newNode(6);

root->right = newNode(9);
```

```
root->left->left = newNode(5);

root->left->right = newNode(3);

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

inorder(root);

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

preorder(root);

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

postorder(root);

getchar();

return 0;

}
```

Question-2

```
#include <stdio.h>

#include <stdlib.h>

struct btnode

{

int value;

struct btnode *leaf;

struct btnode *r;

}*root = NULL, *temp = NULL, *t2, *t1;

int insert();
int inorder(struct btnode *t);

int flag = 1;

int main()

{

int x;

printf("******MENU*******");

printf("\n1 - Insert an element into tree\n2 - Inorder Traversal\n3 - Exit\n");

while(1)

{

printf("\nEnter your choice : ");

scanf("%d", &x);

if(x==1)

{

insert();

}

else if(x==2)

{

inorder(root);

}

else if(x==3)

{

exit(0);

}

else

{
```

```
printf("Wrong choice, Please enter correct choice ");

break;

}

}

return 0;

}

int insert()

{

create();

if (root == NULL)

root = temp;

else

search(root);

return 0;

}

int create()

{

int data;

printf("Enter data of node to be inserted : ");

scanf("%d", &data);

temp = (struct btnode *)malloc(1*sizeof(struct btnode));

temp->value = data;

temp->leaf = temp->r = NULL;

return 0;

}
```

```
int search(struct btnode *t)

{

if ((temp->value > t->value) && (t->r != NULL))

search(t->r);

else if ((temp->value > t->value) && (t->r == NULL))

t->r = temp;

else if ((temp->value < t->value) && (t->leaf != NULL))

search(t->leaf);

else if ((temp->value < t->value) && (t->leaf == NULL))

t->leaf = temp;

return 0;

}

int inorder(struct btnode *t)

{

if (root == NULL)

{

printf("No elements in a tree to display");

return;

}

if (t->leaf != NULL)

inorder(t->leaf);

printf("%d -> ", t->value);

if (t->r != NULL)

inorder(t->r);

return 0;
}
```

Question-3

```
#include <stdio.h>
int main()
{
int arr[300], elements, i, n;
printf("Enter number of elements in array\n");
scanf("%d", &n);
printf("Enter the elements:\n");
for (i = 0; i < n; i++)
{
scanf("%d", &arr[i]);
}
printf("Enter an element to search\n");
scanf("%d", &elements);
for (i = 0; i < n; i++)
{
if (arr[i] == elements)
{
printf("%d is there in the array and at location %d.\n", elements, i+1);
break;
}
}
if (i == n)
{
printf("%d isn't there in the in the array.\n", elements);
}
return 0;
}
```

Question-4

```
#include<stdio.h>
int main ()
{
int arr[300], n, k, i, j, x, small, high, coro, con=0;
printf ("Enter no. of elements in the array\n");
scanf ("%d", &n);
printf("Enter the elements:\n");
for (i = 0; i < n; i++)
{
scanf ("%d ", &arr[i]);
}
printf ("enter the element to search:");
scanf ("%d", &k);
for (i = 0; i < n; i++)
{
```

```
for (j = i + 1; j < n; j++)

{

if (arr[i] > arr[j])

{

x = arr[i];

arr[i] = arr[j];

arr[j] = x;

}

}

}

small = 0;

high = n - 1;

while (small <= high)

{

coro = (small + high) / 2;

if (k == arr[coro])

{

con = 1;

break;

}

else if (k < arr[coro])

{

high = coro - 1;

}

else
```

```
{

small = coro + 1;

}

}

if (con == 0)

{

printf ("%d value not found\n", k);

}

else

{

printf ("%d value found at %d position\n", k,coro + 1);

}

return 0;

}
```