8A. Write a program a) ToconstructabinarySearchtree.

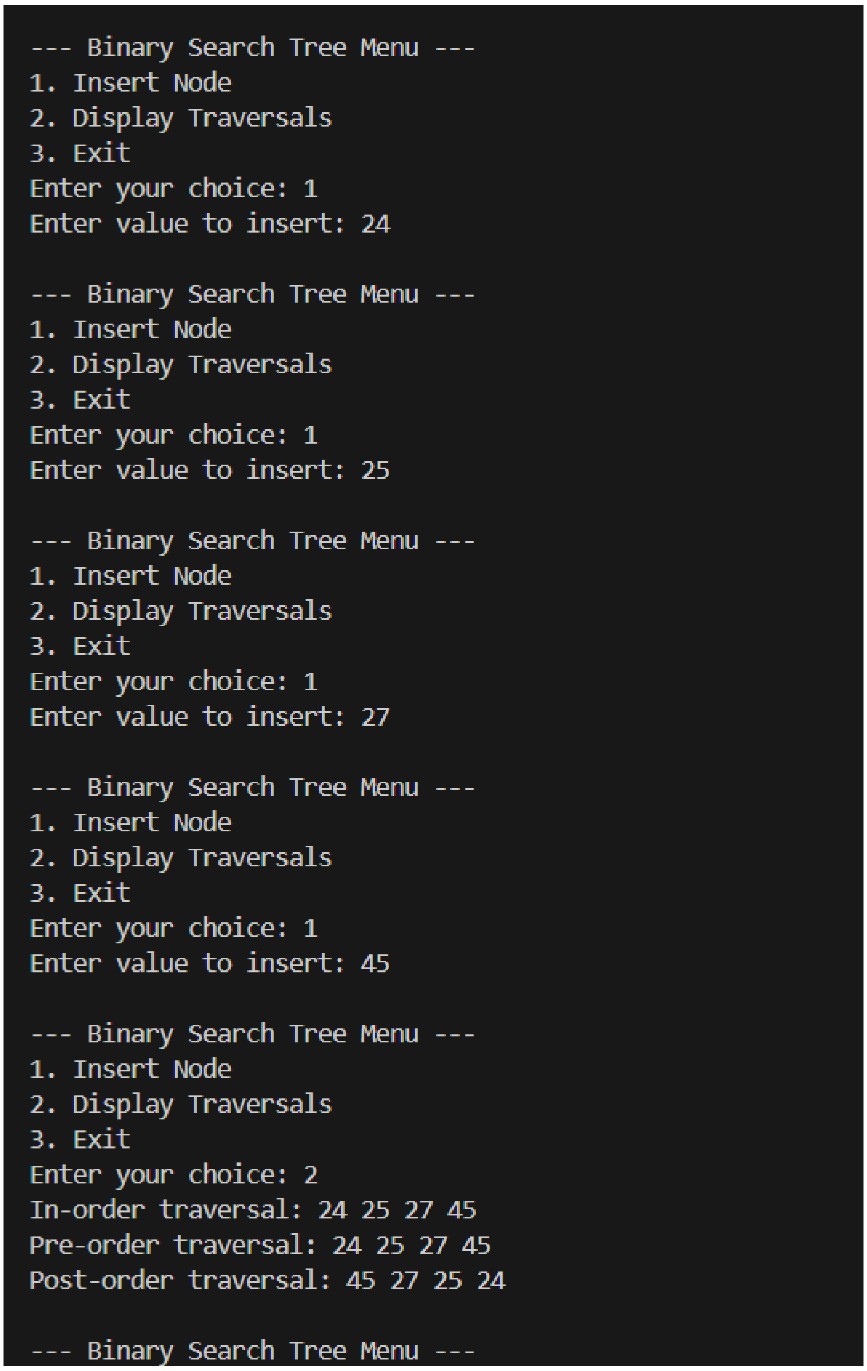
b) To traverse the tree using all the methods i.e., in-order, preorder and post order

c) To display the elements in the tree.

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

// Definition of BST node
struct node {
   int data;
   struct node *left;
   struct node *right;
};

// Function to create a new node
struct node* createNode(int value) {
   struct node *newnode = (struct node*)malloc(sizeof(struct node));
   newnode->data = value;
   newnode->left = newnode->right = NULL;
   return newnode;
}

// Insert node into BST
struct node* insert(struct node* root, int value) {
   if (root == NULL) {
      return createNode(value);
   }
```

```
    if (value < root->data) {
        root->left = insert(root->left, value);
    } else if (value > root->data) {
        root->right = insert(root->right, value);
    }
    return root;
}

// In-order traversal (Left, Root, Right)
void inorder(struct node* root) {
    if (root != NULL) {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
}

// Pre-order traversal (Root, Left, Right)
void preorder(struct node* root) {
    if (root != NULL) {
        printf("%d ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}

// Post-order traversal (Left, Right, Root)
void postorder(struct node* root) {
    if (root != NULL) {
        postorder(root->left);
        postorder(root->right);
        printf("%d ", root->data);
    }
}

// Display BST elements using all traversals
void display(struct node* root) {
    printf("In-order traversal: ");
    inorder(root);
    printf("\n");

    printf("Pre-order traversal: ");
    preorder(root);
    printf("\n");

    printf("Post-order traversal: ");
    postorder(root);
    printf("\n");
}

// Main function
int main() {
    struct node* root = NULL;
    int choice, value;

    do {
        printf("\n--- Binary Search Tree Menu ---\n");
        printf("1. Insert Node\n");
        printf("2. Display Traversals\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
    switch(choice) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                root = insert(root, value);
                break;
            case 2:
                if (root == NULL)
                    printf("BST is empty.\n");
                else
                    display(root);
                break;
            case 3:
                printf("Exiting program.\n");
                break;
            default:
                printf("Invalid choice! Try again.\n");
        }
    } while (choice != 3);

    return 0;
}
```

OUTPUT:



```
--- Binary Search Tree Menu ---
1. Insert Node
2. Display Traversals
3. Exit
Enter your choice: 1
Enter value to insert: 24

--- Binary Search Tree Menu ---
1. Insert Node
2. Display Traversals
3. Exit
Enter your choice: 1
Enter value to insert: 25

--- Binary Search Tree Menu ---
1. Insert Node
2. Display Traversals
3. Exit
Enter your choice: 1
Enter value to insert: 27

--- Binary Search Tree Menu ---
1. Insert Node
2. Display Traversals
3. Exit
Enter your choice: 1
Enter value to insert: 45

--- Binary Search Tree Menu ---
1. Insert Node
2. Display Traversals
3. Exit
Enter your choice: 2
In-order traversal: 24 25 27 45
Pre-order traversal: 24 25 27 45
Post-order traversal: 45 27 25 24

--- Binary Search Tree Menu ---
```