Write a program

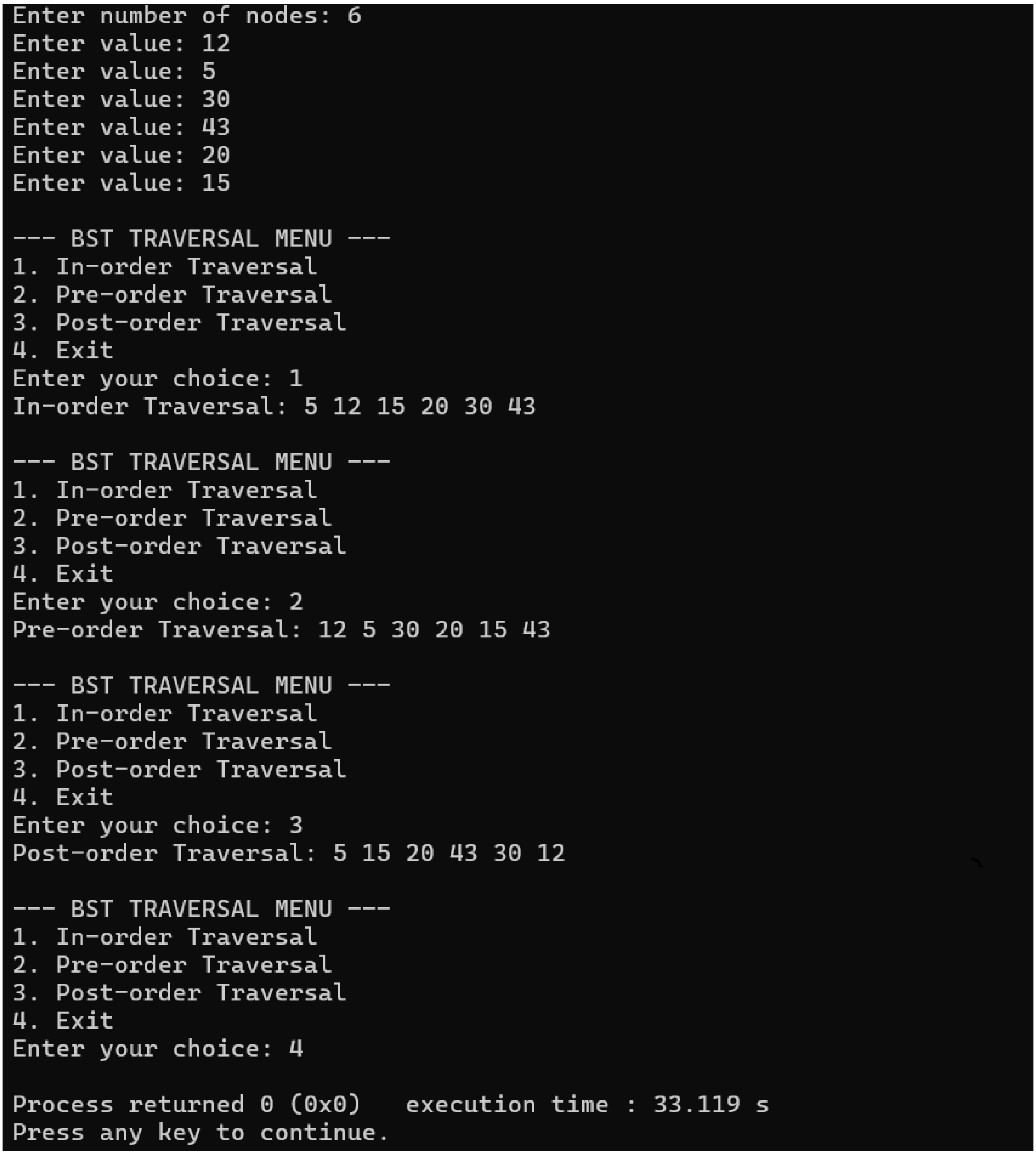
a) To construct a binary Search tree.

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

c) To display the elements in the 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;
   newnode = (struct node *)malloc(sizeof(struct node));
   newnode->data = value;
   newnode->left = NULL;
   newnode->right = NULL;
   return newnode;
}
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;
}
void inorder(struct node *root) {
   if (root != NULL) {
      inorder(root->left);
```

```
        printf("%d ", root->data);
        inorder(root->right);
    }
}
void preorder(struct node *root) {
    if (root != NULL) {
        printf("%d ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
void postorder(struct node *root) {
    if (root != NULL) {
        postorder(root->left);
        postorder(root->right);
        printf("%d ", root->data);
    }
}
int main() {
    struct node *root = NULL;
    int n, val, i, choice;
    printf("Enter number of nodes: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("Enter value: ");
        scanf("%d", &val);
        root = insert(root, val);
    }
    do {
        printf("\n--- BST TRAVERSAL MENU ---\n");
        printf("1. In-order Traversal\n");
        printf("2. Pre-order Traversal\n");
        printf("3. Post-order Traversal\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("In-order Traversal: ");
                inorder(root);
                printf("\n");
                break;
            case 2:
                printf("Pre-order Traversal: ");
                preorder(root);
                printf("\n");
                break;
            case 3:
                printf("Post-order Traversal: ");
                postorder(root);
                printf("\n");
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice\n");
        }
    } while (1);
    return 0;
}
```



```
Enter number of nodes: 6
Enter value: 12
Enter value: 5
Enter value: 30
Enter value: 43
Enter value: 20
Enter value: 15

--- BST TRAVERSAL MENU ---
1. In-order Traversal
2. Pre-order Traversal
3. Post-order Traversal
4. Exit
Enter your choice: 1
In-order Traversal: 5 12 15 20 30 43

--- BST TRAVERSAL MENU ---
1. In-order Traversal
2. Pre-order Traversal
3. Post-order Traversal
4. Exit
Enter your choice: 2
Pre-order Traversal: 12 5 30 20 15 43

--- BST TRAVERSAL MENU ---
1. In-order Traversal
2. Pre-order Traversal
3. Post-order Traversal
4. Exit
Enter your choice: 3
Post-order Traversal: 5 15 20 43 30 12

--- BST TRAVERSAL MENU ---
1. In-order Traversal
2. Pre-order Traversal
3. Post-order Traversal
4. Exit
Enter your choice: 4

Process returned 0 (0x0)   execution time : 33.119 s
Press any key to continue.
```