**WEEK 8**

**2.BINARY SEARCH TREE:**

#include <stdio.h>

#include <stdlib.h>

typedef struct Node {

int data;

struct Node\* left;

struct Node\* right;

} Node;

Node\* createNode(int data) {

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode->data = data;

newNode->left = NULL;

newNode->right = NULL;

return newNode;

}

Node\* insert(Node\* root, int data) {

if (root == NULL) {

return createNode(data);

}

if (data < root->data) {

root->left = insert(root->left, data);

} else if (data > root->data) {

root->right = insert(root->right, data);

}

return root;

}

void inorder(Node\* root) {

if (root != NULL) {

inorder(root->left);

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

inorder(root->right);

}

}

void preorder(Node\* root) {

if (root != NULL) {

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

preorder(root->left);

preorder(root->right);

}

}

void postorder(Node\* root) {

if (root != NULL) {

postorder(root->left);

postorder(root->right);

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

}

}

void display(Node\* root) {

printf("Tree elements (In-order): ");

inorder(root);

printf("\n");

}

int main() {

Node\* root = NULL;

int choice, data;

while (1) {

printf("\nBinary Search Tree Operations:\n");

printf("1. Insert\n");

printf("2. In-order Traversal\n");

printf("3. Pre-order Traversal\n");

printf("4. Post-order Traversal\n");

printf("5. Display Tree Elements\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter data to insert: ");

scanf("%d", &data);

root = insert(root, data);

break;

case 2:

printf("In-order Traversal: ");

inorder(root);

printf("\n");

break;

case 3:

printf("Pre-order Traversal: ");

preorder(root);

printf("\n");

break;

case 4:

printf("Post-order Traversal: ");

postorder(root);

printf("\n");

break;

case 5:

display(root);

break;

case 6:

printf("Exiting...\n");

exit(0);

default:

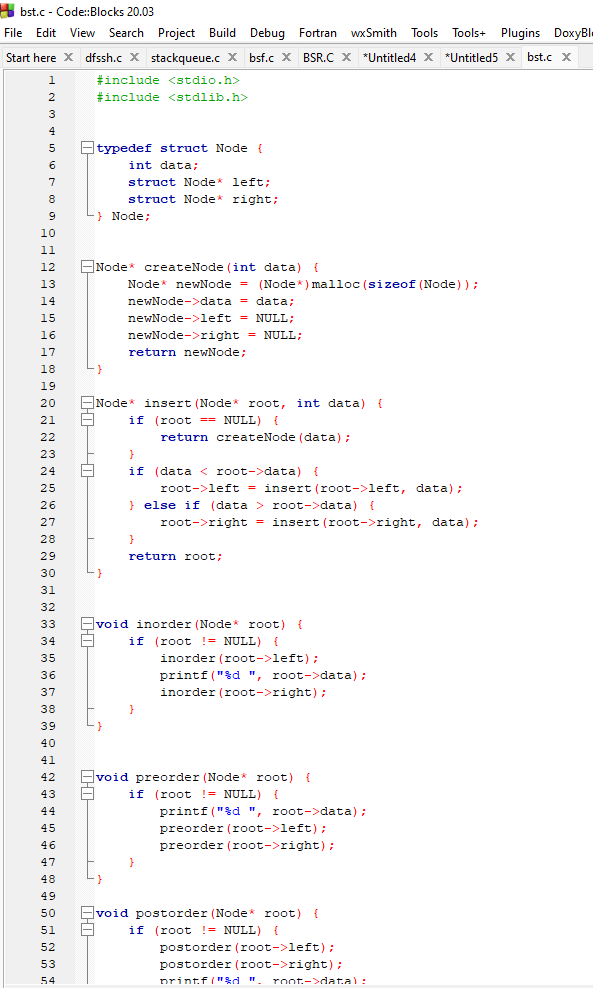
printf("Invalid choice. Please try again.\n");

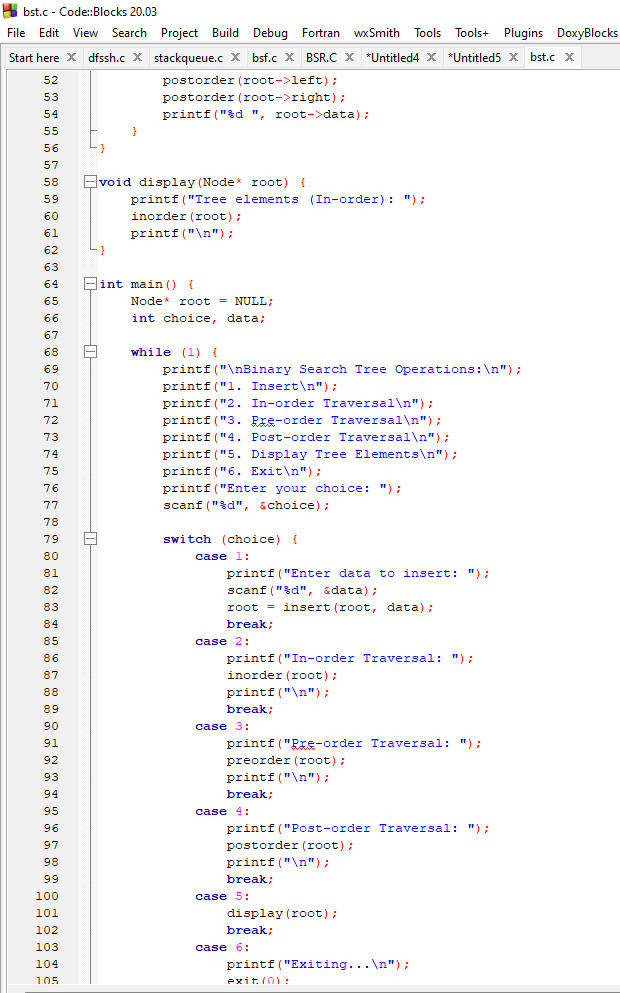
}

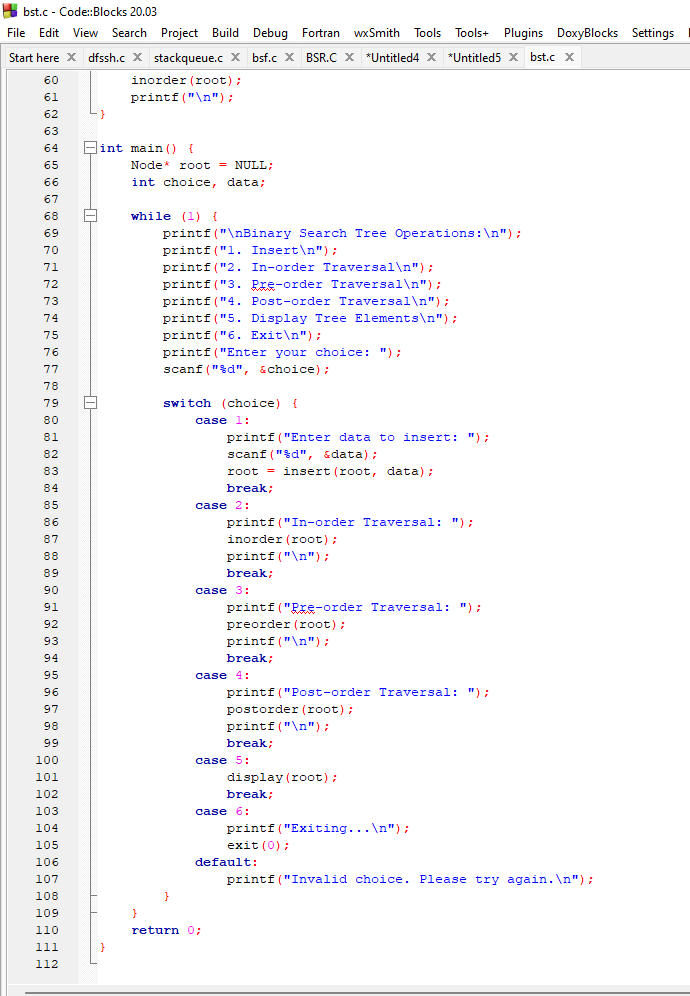
}

return 0;

}

a

a

a

OUTPUT:

