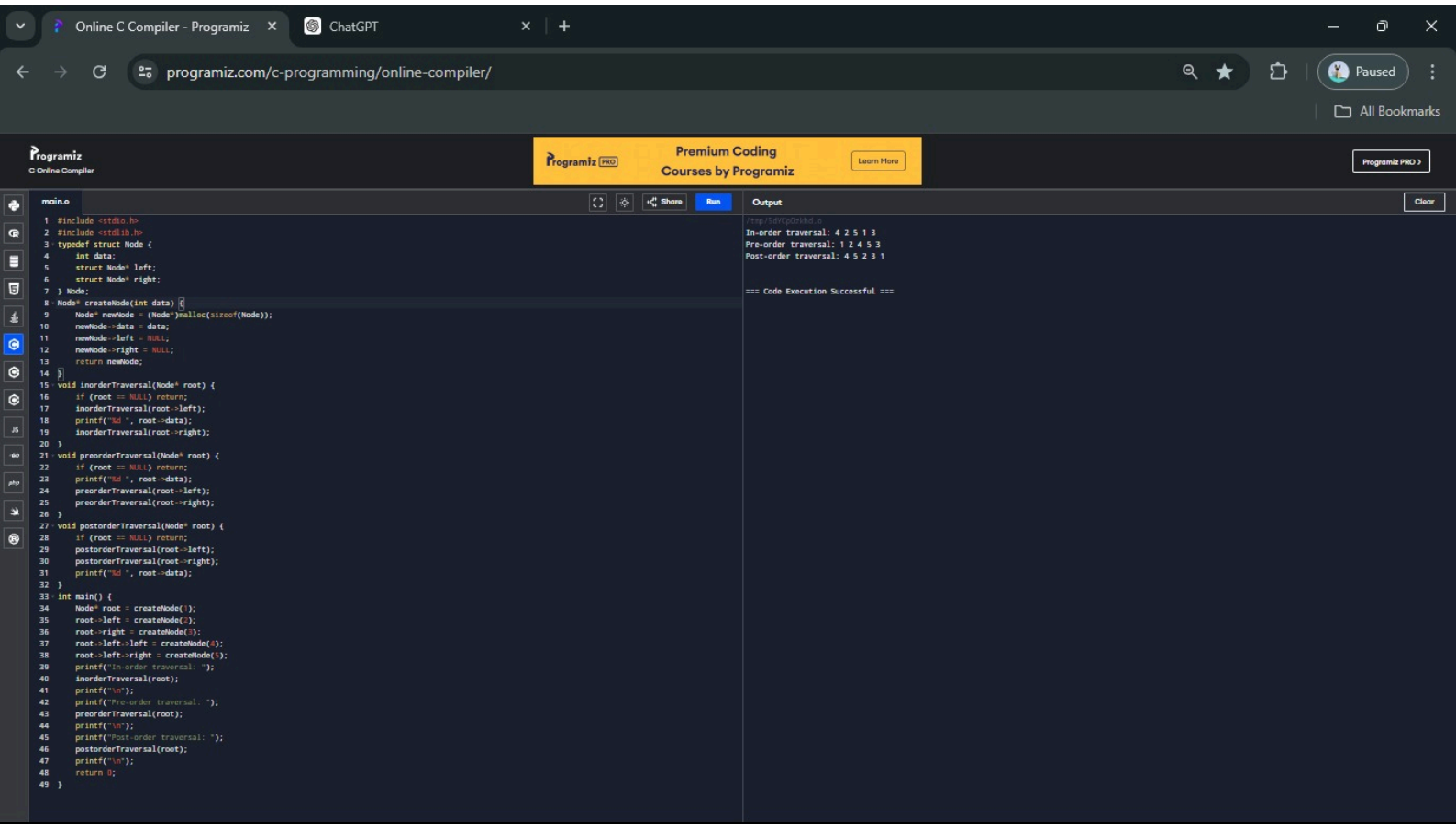
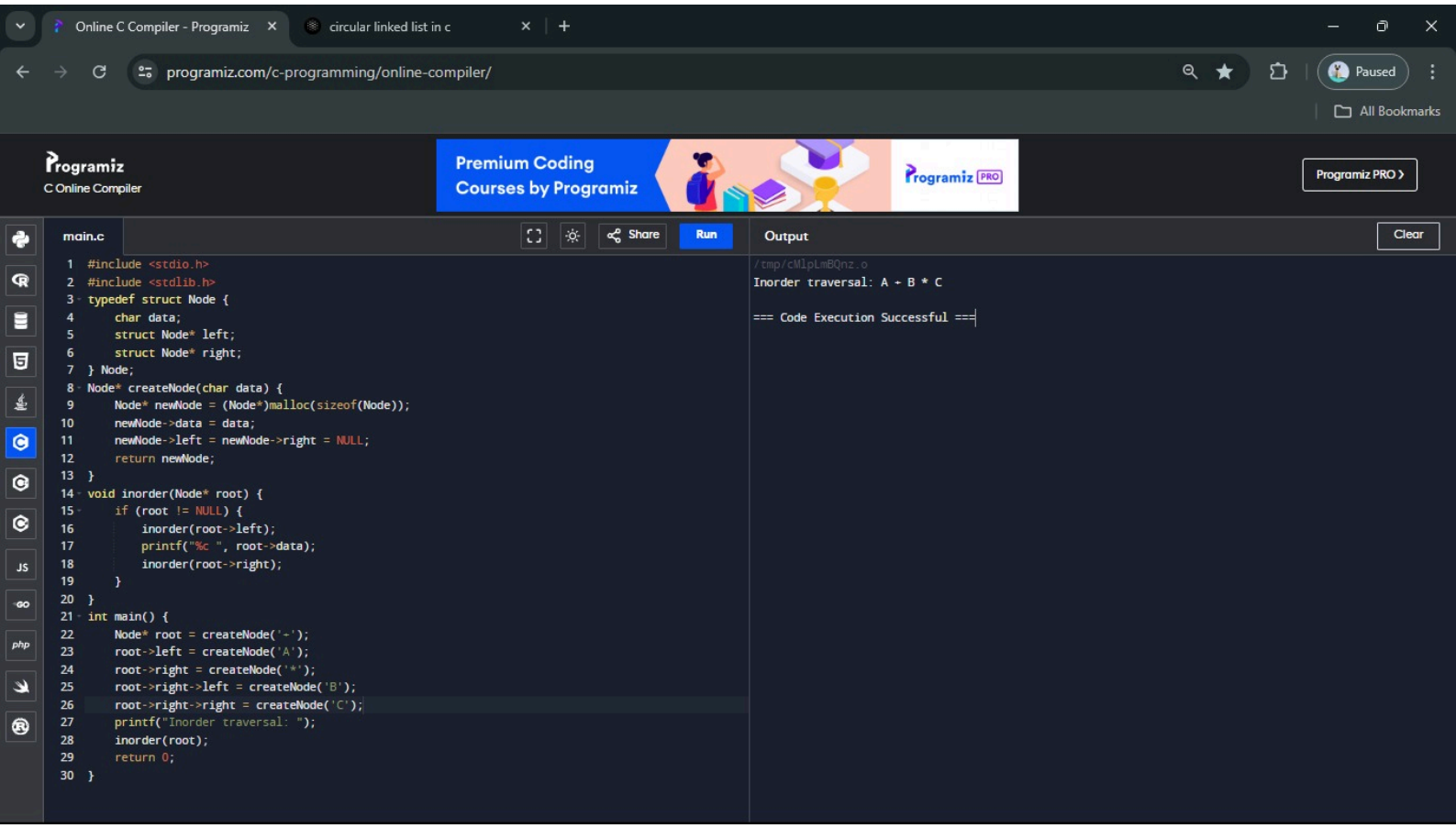
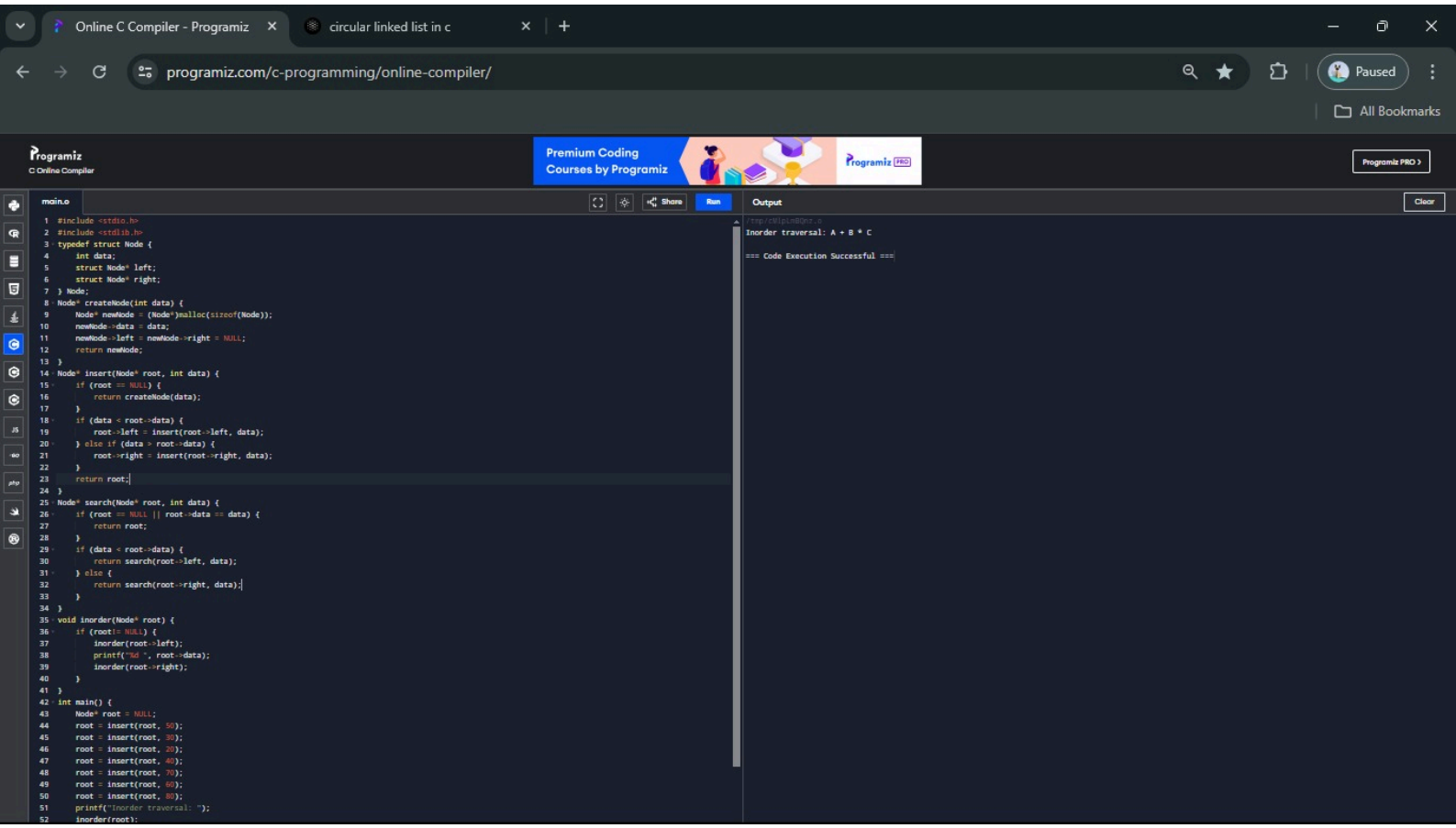

Online C Compiler - Programiz
ChatGPT
programiz.com/c-programming/online-compiler/
Paused
All Bookmarks
Programiz
C Online Compiler
Programiz PRO
Premium Coding
Courses by Programiz
Learn More
Programiz PRO
main.o
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
int 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;
}
void inorderTraversal(Node* root) {
if (root == NULL) return;
inorderTraversal(root->left);
printf("%d ", root->data);
inorderTraversal(root->right);
}
void preorderTraversal(Node* root) {
if (root == NULL) return;
printf("%d ", root->data);
preorderTraversal(root->left);
preorderTraversal(root->right);
}
void postorderTraversal(Node* root) {
if (root == NULL) return;
postorderTraversal(root->left);
postorderTraversal(root->right);
printf("%d ", root->data);
}
int main() {
Node* root = createNode(1);
root->left = createNode(2);
root->right = createNode(3);
root->left->left = createNode(4);
root->left->right = createNode(5);
printf("In-order traversal: ");
inorderTraversal(root);
printf("\n");
printf("Pre-order traversal: ");
preorderTraversal(root);
printf("\n");
printf("Post-order traversal: ");
postorderTraversal(root);
printf("\n");
return 0;
}
In-order traversal: 4 2 5 1 3
Pre-order traversal: 1 2 4 5 3
Post-order traversal: 4 5 2 3 1
=== Code Execution Successful ===


Online C Compiler - Programiz
circular linked list in c
programiz.com/c-programming/online-compiler/
Paused
All Bookmarks
Programiz
C Online Compiler
Premium Coding
Courses by Programiz
Programiz PRO
Programiz PRO ›
main.c
Share
Run
Output
Clear
/tmp/cMlpLmBQnz.o
Inorder traversal: A + B * C
=== Code Execution Successful ===
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
char data;
struct Node* left;
struct Node* right;
} Node;
Node* createNode(char data) {
Node* newNode = (Node*)malloc(sizeof(Node));
newNode->data = data;
newNode->left = newNode->right = NULL;
return newNode;
}
void inorder(Node* root) {
if (root != NULL) {
inorder(root->left);
printf("%c ", root->data);
inorder(root->right);
}
}
int main() {
Node* root = createNode('+');
root->left = createNode('A');
root->right = createNode('*');
root->right->left = createNode('B');
root->right->right = createNode('C');
printf("Inorder traversal: ");
inorder(root);
return 0;
}


Online C Compiler - Programiz
circular linked list in c
programiz.com/c-programming/online-compiler/
Paused
All Bookmarks
Programiz
C Online Compiler
Premium Coding Courses by Programiz
Programiz PRO
Programiz PRO ›
main.o
Share
Run
Output
Clear
#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 = 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;
}
Node* search(Node* root, int data) {
if (root == NULL || root->data == data) {
return root;
}
if (data < root->data) {
return search(root->left, data);
} else {
return search(root->right, data);
}
}
void inorder(Node* root) {
if (root!= NULL) {
inorder(root->left);
printf("%d ", root->data);
inorder(root->right);
}
}
int main() {
Node* root = NULL;
root = insert(root, 50);
root = insert(root, 30);
root = insert(root, 20);
root = insert(root, 40);
root = insert(root, 70);
root = insert(root, 60);
root = insert(root, 80);
printf("Inorder traversal: ");
inorder(root);
Inorder traversal: A + B * C
=== Code Execution Successful ===