Online C Compiler - Programiz

trie in data structure using c lan

programiz.com/c-programming/online-compiler/

Paused

All Bookmarks

Programiz

C Online Compiler

Programiz PRO >

main.c

Share

Run

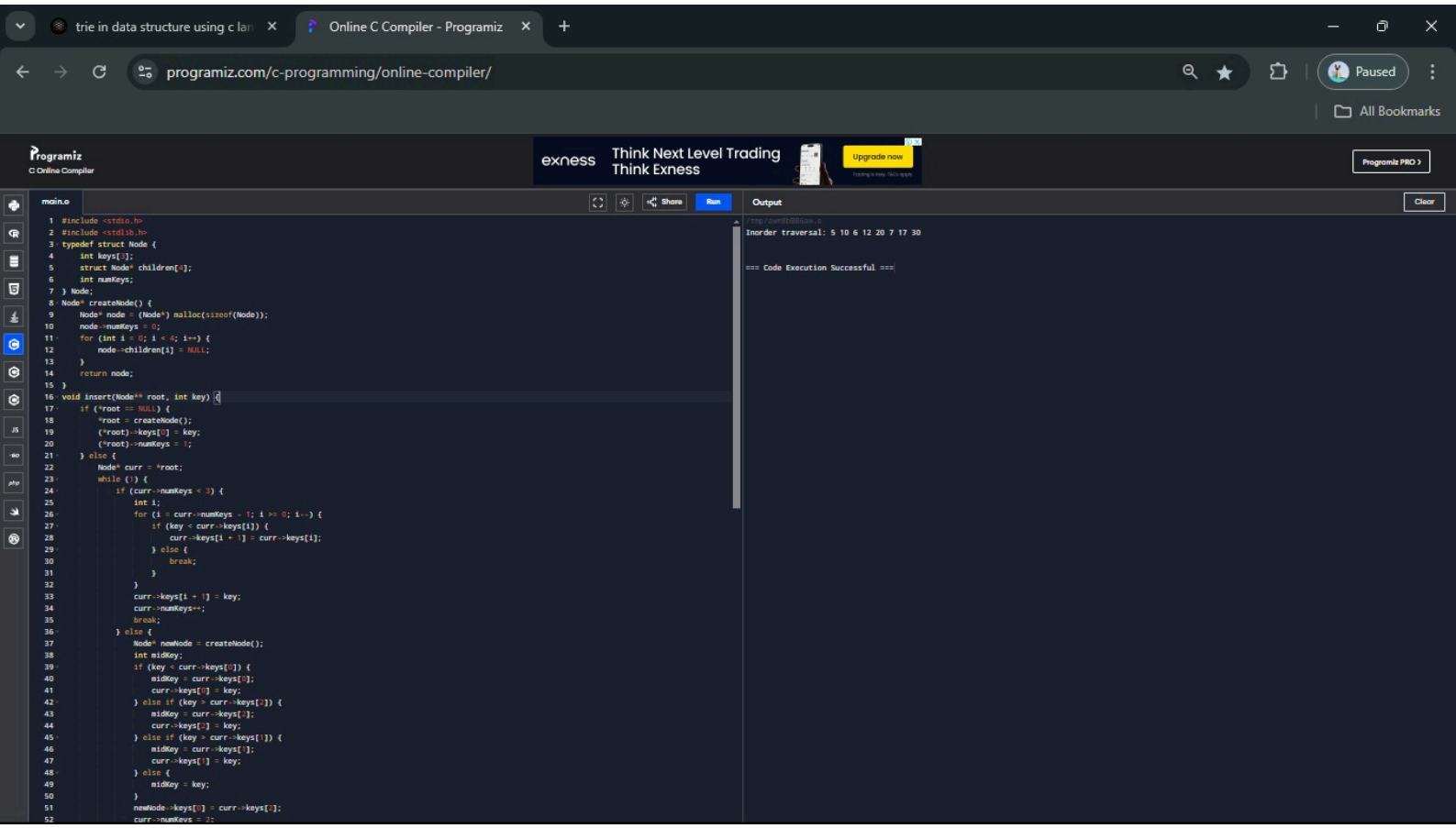
```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int key[2];
    struct Node* child[3];
    int numKeys;
} Node;
Node* createNode() {
    Node* node = (Node*) malloc(sizeof(Node));
    node->numKeys = 0;
    for (int i = 0; i < 3; i++) {
        node->child[i] = NULL;
    }
    return node;
}
void insert(Node** root, int key) {
    if (*root == NULL) {
        *root = createNode();
        (*root)->key[0] = key;
        (*root)->numKeys = 1;
    } else {
        Node* curr = *root;
        while (1) {
            if (curr->numKeys < 2) {
                int i;
                for (i = curr->numKeys - 1; i >= 0; i--) {
                    if (key < curr->key[i]) {
                        curr->key[i + 1] = curr->key[i];
                    } else {
                        break;
                    }
                }
                curr->key[i + 1] = key;
                curr->numKeys++;
                break;
            } else {
                Node* newNode = createNode();
                int midKey;
                if (key < curr->key[0]) {
                    midKey = curr->key[0];
                    curr->key[0] = key;
                } else if (key > curr->key[1]) {
                    midKey = curr->key[1];
                    curr->key[1] = key;
                } else {
                    midKey = key;
                }
                newNode->key[0] = curr->key[1];
                curr->numKeys = 1;
                newNode->numKeys = 1;
                if (curr->child[0] != NULL) {
                    newNode->child[0] = curr->child[2];
```

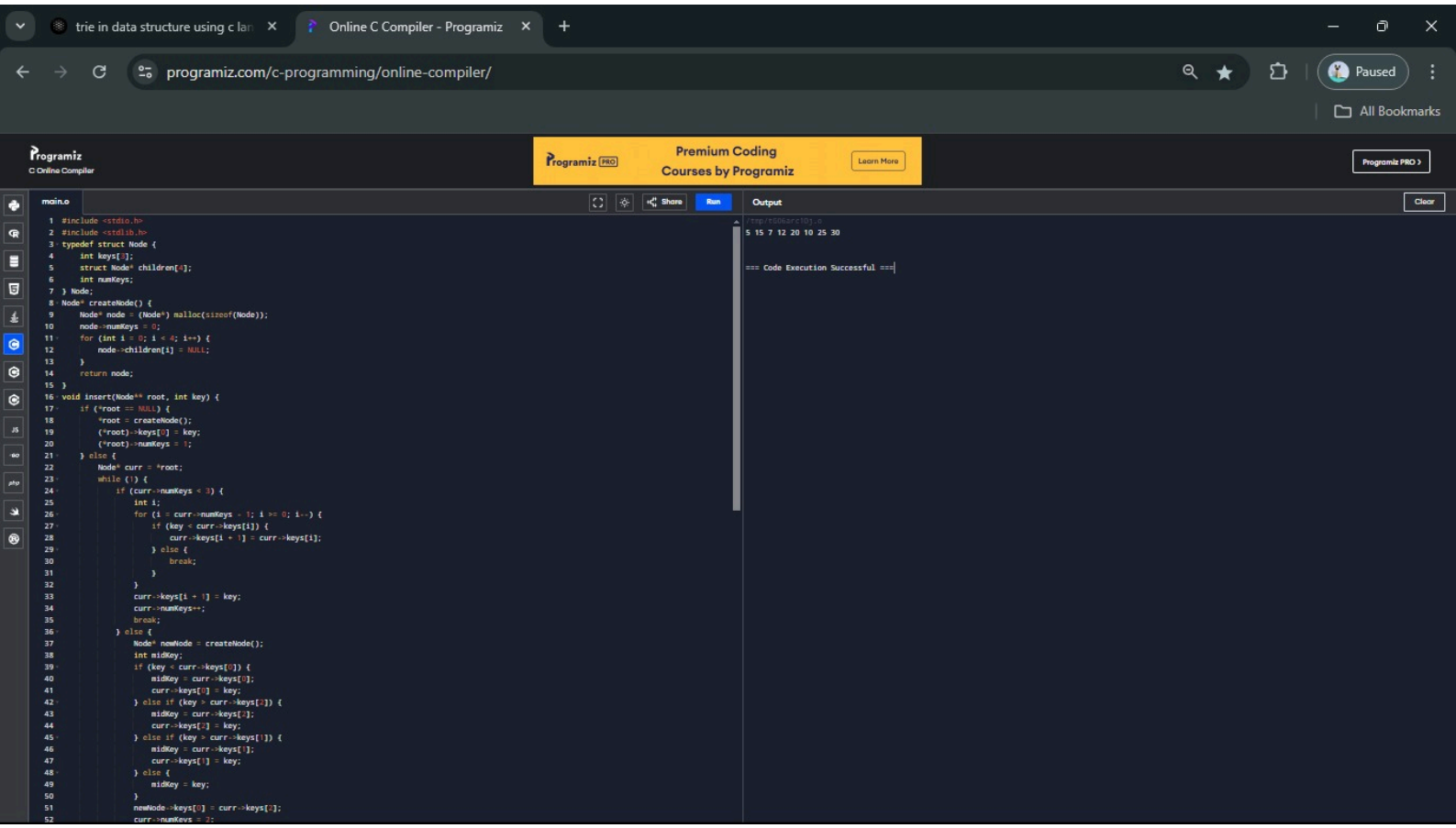
Output

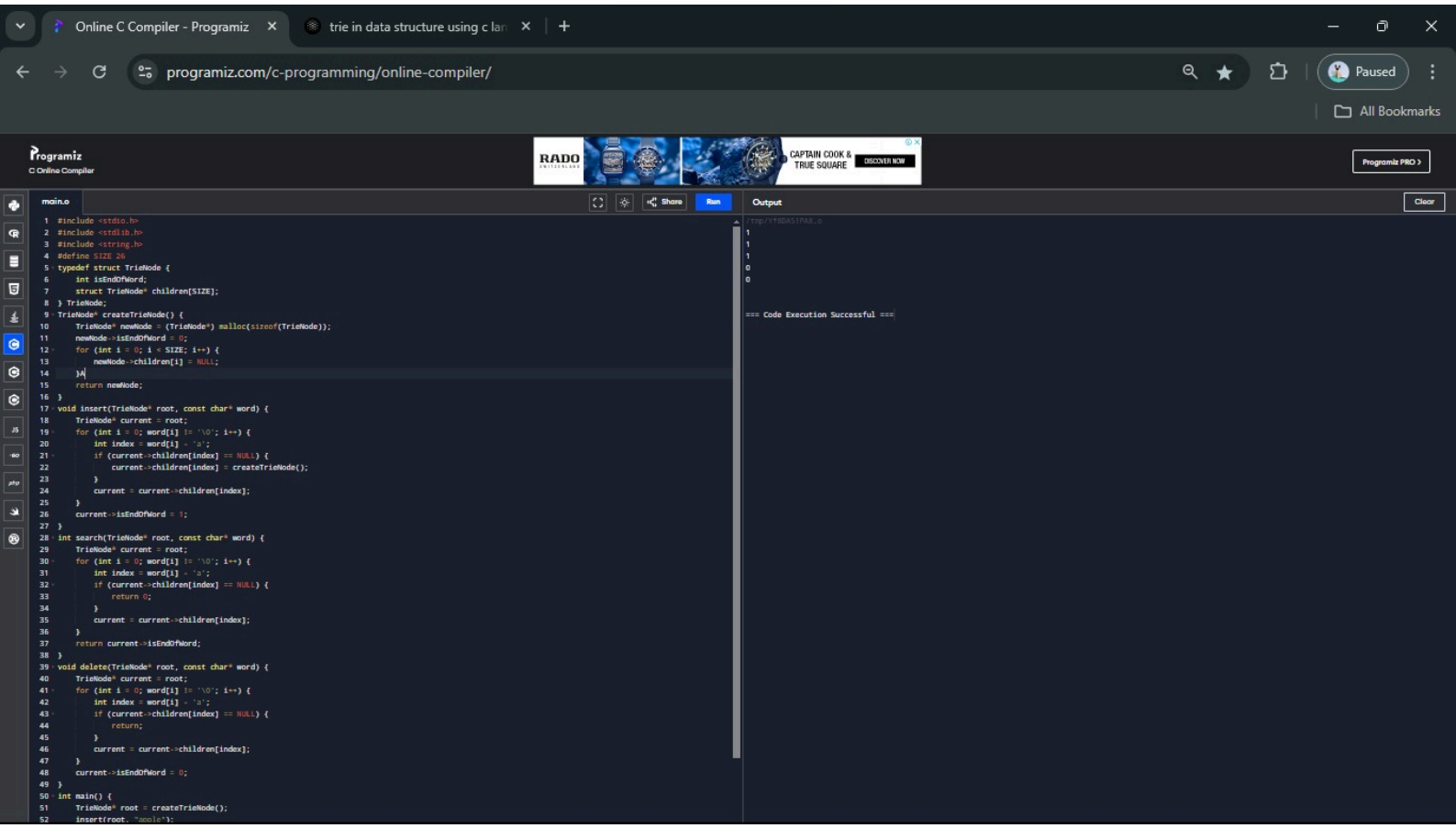
Clear

```
/tmp/g3xhGhKoqn.o
Inorder traversal: 5 6 20 7 12 10 17 30


=== Code Execution Successful ===
```


trie in data structure using c lan
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Paused
All Bookmarks
Programiz
C Online Compiler
exness
Think Next Level Trading
Think Exness
Upgrade now
Programiz PRO
main.o
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
int keys[3];
struct Node* children[4];
int numKeys;
} Node;
Node* createNode() {
Node* node = (Node*) malloc(sizeof(Node));
node->numKeys = 0;
for (int i = 0; i < 4; i++) {
node->children[i] = NULL;
}
return node;
}
void insert(Node** root, int key) {
if (*root == NULL) {
*root = createNode();
(*root)->keys[0] = key;
(*root)->numKeys = 1;
} else {
Node* curr = *root;
while (1) {
if (curr->numKeys < 3) {
int i;
for (i = curr->numKeys - 1; i >= 0; i--) {
if (key < curr->keys[i]) {
curr->keys[i + 1] = curr->keys[i];
} else {
break;
}
}
curr->keys[i + 1] = key;
curr->numKeys++;
break;
} else {
Node* newNode = createNode();
int midKey;
if (key < curr->keys[0]) {
midKey = curr->keys[0];
curr->keys[0] = key;
} else if (key > curr->keys[2]) {
midKey = curr->keys[2];
curr->keys[2] = key;
} else if (key > curr->keys[1]) {
midKey = curr->keys[1];
curr->keys[1] = key;
} else {
midKey = key;
}
newNode->keys[0] = curr->keys[2];
curr->numKeys = 2;
Inorder traversal: 5 10 6 12 20 7 17 30
=== Code Execution Successful ===


trie in data structure using c lan
Online C Compiler - Programiz
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 keys[3];
struct Node* children[4];
int numKeys;
} Node;
Node* createNode() {
Node* node = (Node*) malloc(sizeof(Node));
node->numKeys = 0;
for (int i = 0; i < 4; i++) {
node->children[i] = NULL;
}
return node;
}
void insert(Node** root, int key) {
if (*root == NULL) {
*root = createNode();
(*root)->keys[0] = key;
(*root)->numKeys = 1;
} else {
Node* curr = *root;
while (1) {
if (curr->numKeys < 3) {
int i;
for (i = curr->numKeys - 1; i >= 0; i--) {
if (key < curr->keys[i]) {
curr->keys[i + 1] = curr->keys[i];
} else {
break;
}
}
curr->keys[i + 1] = key;
curr->numKeys++;
break;
} else {
Node* newNode = createNode();
int midKey;
if (key < curr->keys[0]) {
midKey = curr->keys[0];
curr->keys[0] = key;
} else if (key > curr->keys[2]) {
midKey = curr->keys[2];
curr->keys[2] = key;
} else if (key > curr->keys[1]) {
midKey = curr->keys[1];
curr->keys[1] = key;
} else {
midKey = key;
}
newNode->keys[0] = curr->keys[2];
curr->numKeys = 2;
5 15 7 12 20 10 25 30
=== Code Execution Successful ===


Online C Compiler - Programiz
trie in data structure using c lan
programiz.com/c-programming/online-compiler/
Paused
All Bookmarks
Programiz
C Online Compiler
RADO
CAPTAIN COOK & TRUE SQUARE
DISCOVER NOW
Programiz PRO
main.o
Share
Run
Output
Clear
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define SIZE 26
typedef struct TrieNode {
int isEndOfWord;
struct TrieNode* children[SIZE];
} TrieNode;
TrieNode* createTrieNode() {
TrieNode* newNode = (TrieNode*) malloc(sizeof(TrieNode));
newNode->isEndOfWord = 0;
for (int i = 0; i < SIZE; i++) {
newNode->children[i] = NULL;
}
return newNode;
}
void insert(TrieNode* root, const char* word) {
TrieNode* current = root;
for (int i = 0; word[i] != '\0'; i++) {
int index = word[i] - 'a';
if (current->children[index] == NULL) {
current->children[index] = createTrieNode();
}
current = current->children[index];
}
current->isEndOfWord = 1;
}
int search(TrieNode* root, const char* word) {
TrieNode* current = root;
for (int i = 0; word[i] != '\0'; i++) {
int index = word[i] - 'a';
if (current->children[index] == NULL) {
return 0;
}
current = current->children[index];
}
return current->isEndOfWord;
}
void delete(TrieNode* root, const char* word) {
TrieNode* current = root;
for (int i = 0; word[i] != '\0'; i++) {
int index = word[i] - 'a';
if (current->children[index] == NULL) {
return;
}
current = current->children[index];
}
current->isEndOfWord = 0;
}
int main() {
TrieNode* root = createTrieNode();
insert(root, "apple");
/tmp/Yf8DA51PA8.o
1
1
1
0
0
=== Code Execution Successful ===