

main.c



Share

Run

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<stdbool.h>
4
5  #define ALPHABET_SIZE 26
6
7  struct TrieNode{
8      struct TrieNode * children[ALPHABET_SIZE];
9      bool isEndOfWord;
10
11 };
12
13 struct TrieNode *createNode(){
14     struct TrieNode *node = (struct TrieNode *) malloc(sizeof(struct TrieNode));
15
16     node->isEndOfWord = false;
17
18     for(int i =0; i < ALPHABET_SIZE; i++)
19         node->children[i] = NULL;
20
21 }
```

main.c

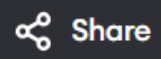


Share

Run

```
22 return node;
23
24 };
25
26 void insert(struct TrieNode *root, const char *word){
27     struct TrieNode *current = root;
28     while(*word){
29         int index = *word - 'a';
30         if(!current->children[index])
31             current->children[index] = createNode();
32         current = current->children[index];
33         word++;
34     }
35
36     current->isEndOfWord= true;
37 }
38
39 bool search (struct TrieNode * root, const char *word){
40     struct TrieNode *current = root;
41
42     while(*word){
```

main.c



Run

```
38
39 bool search (struct TrieNode * root, const char *word){
40     struct TrieNode *current = root;
41
42     while(*word){
43         int index = *word - 'a';
44         if(!current->children[index])
45             return false;
46         current = current->children[index];
47         word++;
48     }
49     return current->isEndOfWord;
50 }
51
52
53
54 int main(){
55     struct TrieNode *root = createNode();
56
57     insert(root, "hello");
58     insert(root, "world");
59     insert(root, "hi");
```

main.c



Share

Run

```
49     return current->isEndOfWord;
50 }
51
52
53
54 int main(){
55     struct TrieNode *root = createNode();
56
57     insert(root, "hello");
58     insert(root, "world");
59     insert(root, "hi");
60     insert(root, "trie");
61
62     printf("search 'hello': %s\n", search (root, "hello") ? "Found" : "Not Found"
        );
63     printf("search 'hi': %s\n", search (root, "hi") ? "Found" : "Not Found");
64     printf("search 'her': %s\n", search (root, "her") ? "Found" : "Not Found");
65     printf("search 'world': %s\n", search (root, "world") ? "Found" : "Not Found"
        );
66     return 0;
67 }
68
```

Output

Clear

```
search 'hello': Found  
search 'hi': Found  
search 'her': Not Found  
search 'world': Found
```

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