

main.c

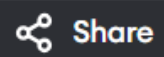


Share

Run

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

main.c



Run

```
21
22 return node;
23
24 };
25 void insert(struct TrieNode *root, const char *word){
26     struct TrieNode *current = root;
27     while(*word){
28         int index = *word - 'a';
29         if(!current -> children[index])
30             current->children[index] = createNode();
31         current = current->children[index];
32         word++;
33     }
34
35     current->isEndOfWord= 1;
36 }
37 int countChildren(struct TrieNode *node, int *index){
38     int count = 0;
39     for(int i =0;i< ALPHABET_SIZE; i++){
40         if(node->children[i]){
41             count++;
42         }
43     }
```

```
36 }
37 int countChildren(struct TrieNode *node, int *index){
38     int count = 0;
39     for(int i = 0; i < ALPHABET_SIZE; i++){
40         if(node->children[i]){
41             count++;
42             *index = i;
43         }
44     }
45     return count;
46 }
47
48 void longestCommonPrefix(struct TrieNode *root, char *prefix){
49     struct TrieNode * current = root;
50     int index;
51     int length = 0;
52
53     while(countChildren(current, &index) == 1 && !current->isEndOfWord){
54         prefix[length++] = 'a' + index;
55         current = current->children[index];
56     }
```

```
57     prefix[length] = '\0';
58 }
59
60 int main(){
61     char *words[] = { "flower", "flow", "flight"};
62     int n = sizeof(words) / sizeof(words[0]);
63
64     struct TrieNode *root = createNode();
65
66     for(int i = 0; i< n;i++)
67         insert(root, words[i]);
68
69     char prefix[100];
70     longestCommonPrefix(root, prefix);
71
72     if(strlen(prefix) > 0){
73         printf("Longest Common Prefix: %s\n",prefix);
74     }
75     else{
76         printf("No common Prefix\n");
77     }
```

## Output

Clear

Longest Common Prefix: fl

=== Code Execution Successful ===