main.c

Share Run

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define ALPHABET_SIZE 26

struct TrieNode{
    struct TrieNode * children[ALPHABET_SIZE];
    int isEndOfWord;
};
struct TrieNode *createNode(){
    struct TrieNode *node = (struct TrieNode *) malloc(sizeof(struct TrieNode));
    node->isEndOfWord = 0;
    for(int i =0; i < ALPHABET_SIZE; i++)
        node->children[i] = NULL;
        return node;

};
void insert(struct TrieNode *root, const char *word){
    struct TrieNode *current = root;
    while(*word){
        int index = *word - 'a';
        if(!current -> children[index])
            current->children[index] = createNode();
        current = current->children[index];
        word++;
```

main.c

Share Run

```
    }
    current->isEndOfWord= 1;
}
int countChildren(struct TrieNode *node, int *index){
    int count = 0;
    for(int i =0;i< ALPHABET_SIZE; i++){
        if(node->children[i]){
            count++;
            *index = i;
        }
    }
    return count;
}

void longestCommonPrefix(struct TrieNode *root, char *prefix){
    struct TrieNode * current = root;
    int index;
    int length = 0;

    while(countChildren(current, &index) == 1 && !current->isEndOfWord){
        prefix[length++] = 'a' + index;
        current =current->children[index];
    }
    prefix[length] = '\0';
}

```

main.c

Share Run

```
    struct TrieNode * current = root;
    int index;
    int length = 0;

    while(countChildren(current, &index) == 1 && !current->isEndOfWord){
        prefix[length++] = 'a' + index;
        current =current->children[index];
    }
    prefix[length] = '\0';
}

int main(){
    char *words[] = { "flower", "flow", "flight"};
    int n = sizeof(words) / sizeof(words[0]);
    struct TrieNode *root = createNode();
    for(int i = 0; i< n;i++)
        insert(root, words[i]);
        char prefix[100];
        longestCommonPrefix(root, prefix);
        if(strlen(prefix) > 0){
            printf("Longest Common Prefix: %s\n",prefix);
        }
        else{
            printf("No common Prefix\n");
        }
}
```

Output

```
Longest Common Prefix: fl


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