Implement a C program to eliminate left factoring.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_PROD 10

#define MAX\_LEN 20

typedef struct {

    char nonTerminal;

    char production[MAX\_LEN];

} Production;

void leftFactoring(Production\* productions, int numProductions) {

    int i, j;

    int commonPrefixLen = 0;

    char commonPrefix[MAX\_LEN] = "";

    for (i = 0; i < numProductions; i++) {

        for (j = i + 1; j < numProductions; j++) {

            int len = 0;

            while (productions[i].production[len] == productions[j].production[len] && productions[i].production[len] != '\0') {

                len++;

            }

            if (len > commonPrefixLen) {

                commonPrefixLen = len;

                strncpy(commonPrefix, productions[i].production, commonPrefixLen);

                commonPrefix[commonPrefixLen] = '\0';

            }

        }

    }

    if (commonPrefixLen > 0) {

        printf("Common prefix found: %s\n", commonPrefix);

        printf("After left factoring:\n");

        for (i = 0; i < numProductions; i++) {

            if (strncmp(productions[i].production, commonPrefix, commonPrefixLen) == 0) {

                printf("%c' -> %s%c'\n", productions[i].nonTerminal, productions[i].production + commonPrefixLen, productions[i].nonTerminal);

            } else {

                printf("%c -> %s%c'\n", productions[i].nonTerminal, productions[i].production, productions[i].nonTerminal);

            }

        }

    } else {

        printf("No common prefix found. Left factoring not applicable.\n");

    }

}

int main() {

    int numProductions;

    Production productions[MAX\_PROD];

    printf("Enter the number of productions: ");

    scanf("%d", &numProductions);

    printf("Enter the productions (e.g., S->abc):\n");

    for (int i = 0; i < numProductions; i++) {

        scanf(" %c -> %s", &productions[i].nonTerminal, productions[i].production);

    }

    leftFactoring(productions, numProductions);

    return 0;

}

Output:

