CODE

#include <stdio.h>

#include <string.h>

#define MAX 100

int commonPrefix(char str1[], char str2[]) {

int i = 0;

while (str1[i] && str2[i] && str1[i] == str2[i]) {

i++;

}

return i;

}

int main() {

char production[MAX], prod1[MAX], prod2[MAX];

char non\_terminal;

int i = 0, j = 0, k = 0;

printf("Enter production (e.g. A=abcd/abef):\n");

scanf("%s", production);

non\_terminal = production[0];

i = 2; // skip "A="

while (production[i] != '/' && production[i] != '\0') {

prod1[j++] = production[i++];

}

prod1[j] = '\0';

i++;

j = 0;

while (production[i] != '\0') {

prod2[j++] = production[i++];

}

prod2[j] = '\0';

int prefixLen = commonPrefix(prod1, prod2);

if (prefixLen > 0) {

printf("\nGrammar needs left factoring.\n");

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

printf("%c -> ", non\_terminal);

for (k = 0; k < prefixLen; k++) {

printf("%c", prod1[k]);

}

printf("%c'\n", non\_terminal);

printf("%c' -> ", non\_terminal);

if (prod1[prefixLen] == '\0')

printf("ε");

else

printf("%s", &prod1[prefixLen]);

printf(" | ");

if (prod2[prefixLen] == '\0')

printf("ε\n");

else

printf("%s\n", &prod2[prefixLen]);

} else {

printf("\nNo left factoring needed.\n");

printf("Production: %s\n", production);

}

return 0;

}

SCREENSHOTS:

