

```
1 #include <stdio.h>
2 #include <string.h>
3 #define SIZE 50
4 int main() {
5     int num;
6     char production[10][SIZE];
7     char non_terminal, alpha, beta;
8     printf("Enter Number of Productions: ");
9     scanf("%d", &num);
10    printf("Enter the grammar :\n");
11    for (int i = 0; i < num; i++) {
12        scanf("%s", production[i]);
13    }
14    for (int i = 0; i < num; i++) {
15        printf("\nGRAMMAR: %s", production[i]);
16        non_terminal = production[i][0];
17        int index = 3;
18        if (production[i][index] == non_terminal) {
19            alpha = production[i][index + 1];
20            printf(" is left recursive.\n");
21            while (production[i][index] != '\0' && production[i][index] != '|')
22                index++;
23            if (production[i][index] != '\0') {
24                beta = production[i][index + 1];
25                printf("Grammar without left recursion:\n");
26                printf("%c->%c%c'\n", non_terminal, beta, non_terminal);
27                printf("%c'->%c%c'|\n", non_terminal, alpha, non_terminal);
28            } else
29                printf(" can't be reduced\n");
30        } else
31            printf(" is not left recursive.\n");
32    }
33    return 0;
34 }
```

Enter Number of Productions: 2

Enter the grammar :

$S \rightarrow (L) | a$

$L \rightarrow L, S | S$

GRAMMAR: $S \rightarrow (L) | a$ is not left recursive.

GRAMMAR: $L \rightarrow L, S | S$ is left recursive.

Grammar without left recursion:

$L \rightarrow SL'$

$L' \rightarrow , L' | \epsilon$

=== Code Execution Successful ===