CSB 353: Compiler Design

LAB 5 - 6

Submitted By:

Name: PREM KUMAR

Roll No: 191210037

Branch: CSE

Semester: 6 th

Submitted To: Dr. Shelly Sachdeva

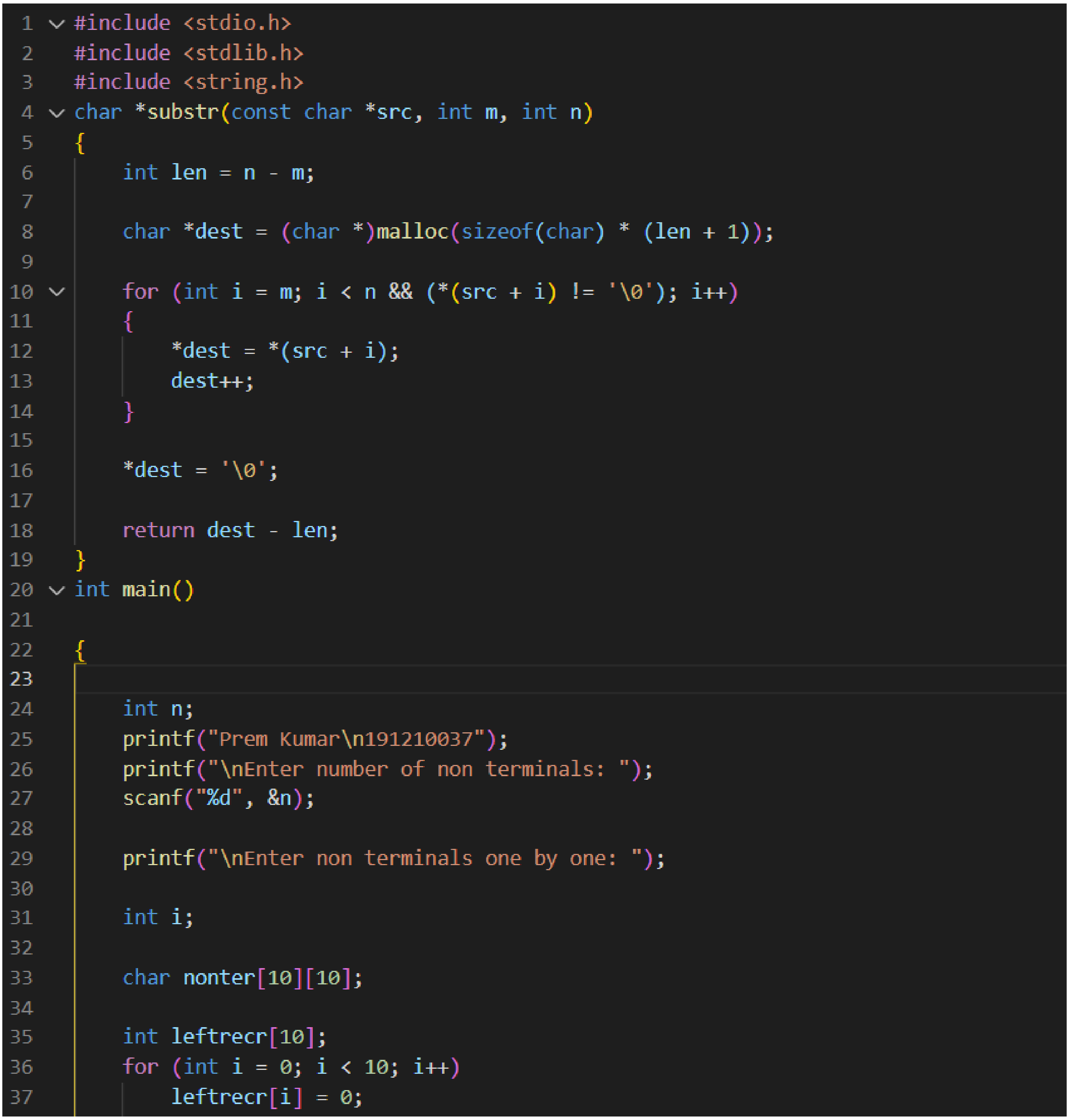
Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

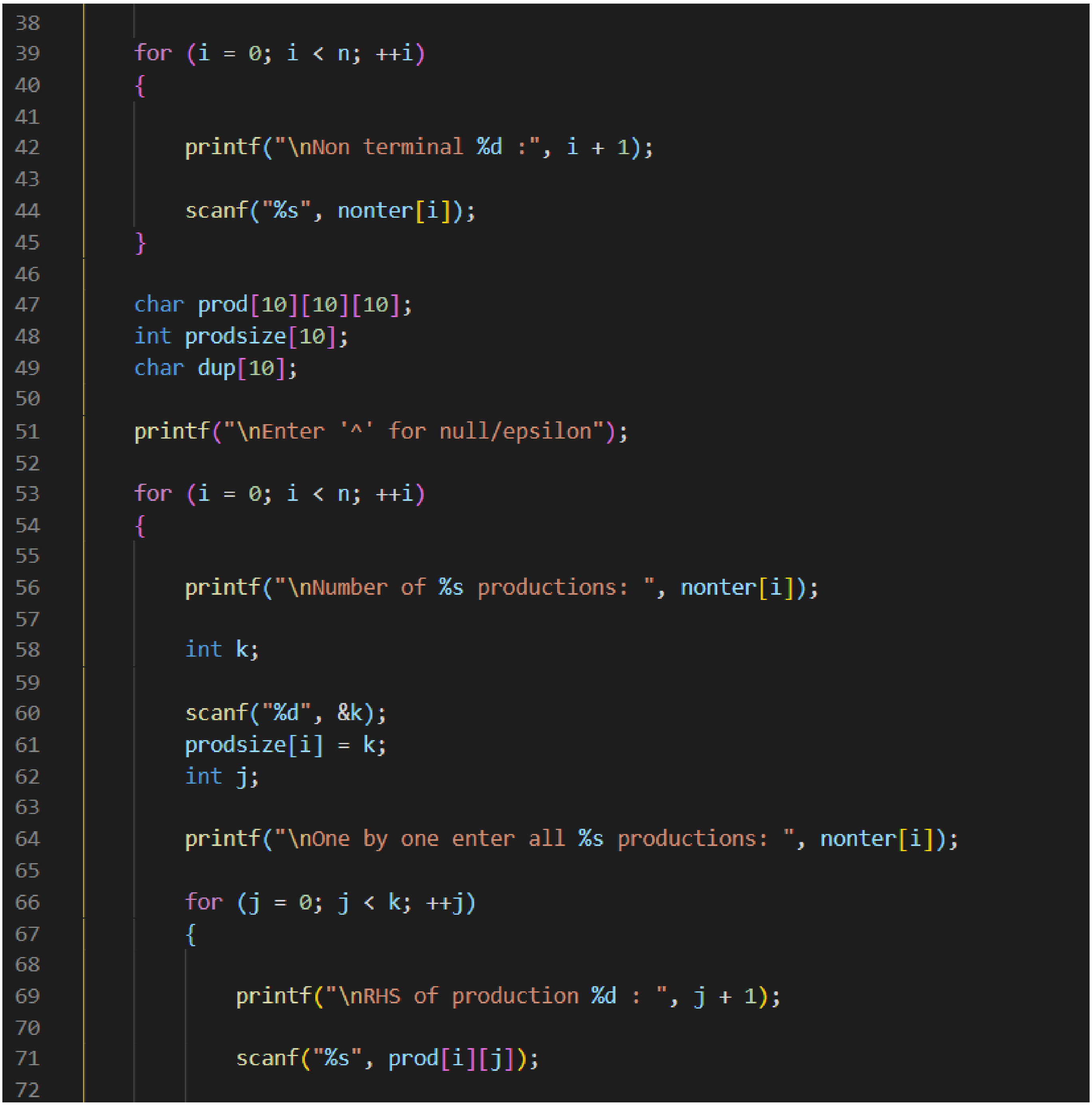
2019-2023

Ques 1. Write a C program that accepts a grammar and removes left recursion from it.

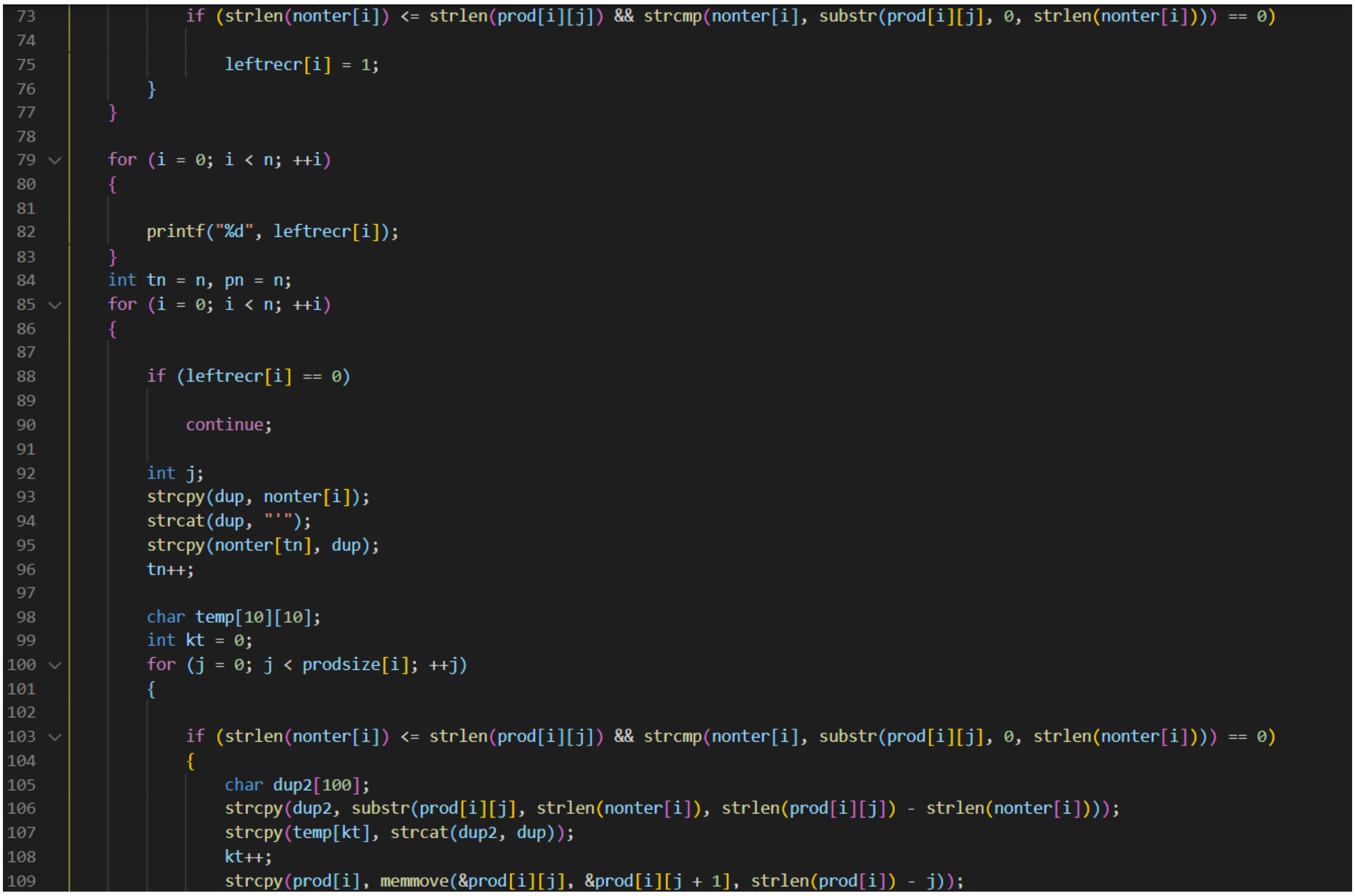
Code:



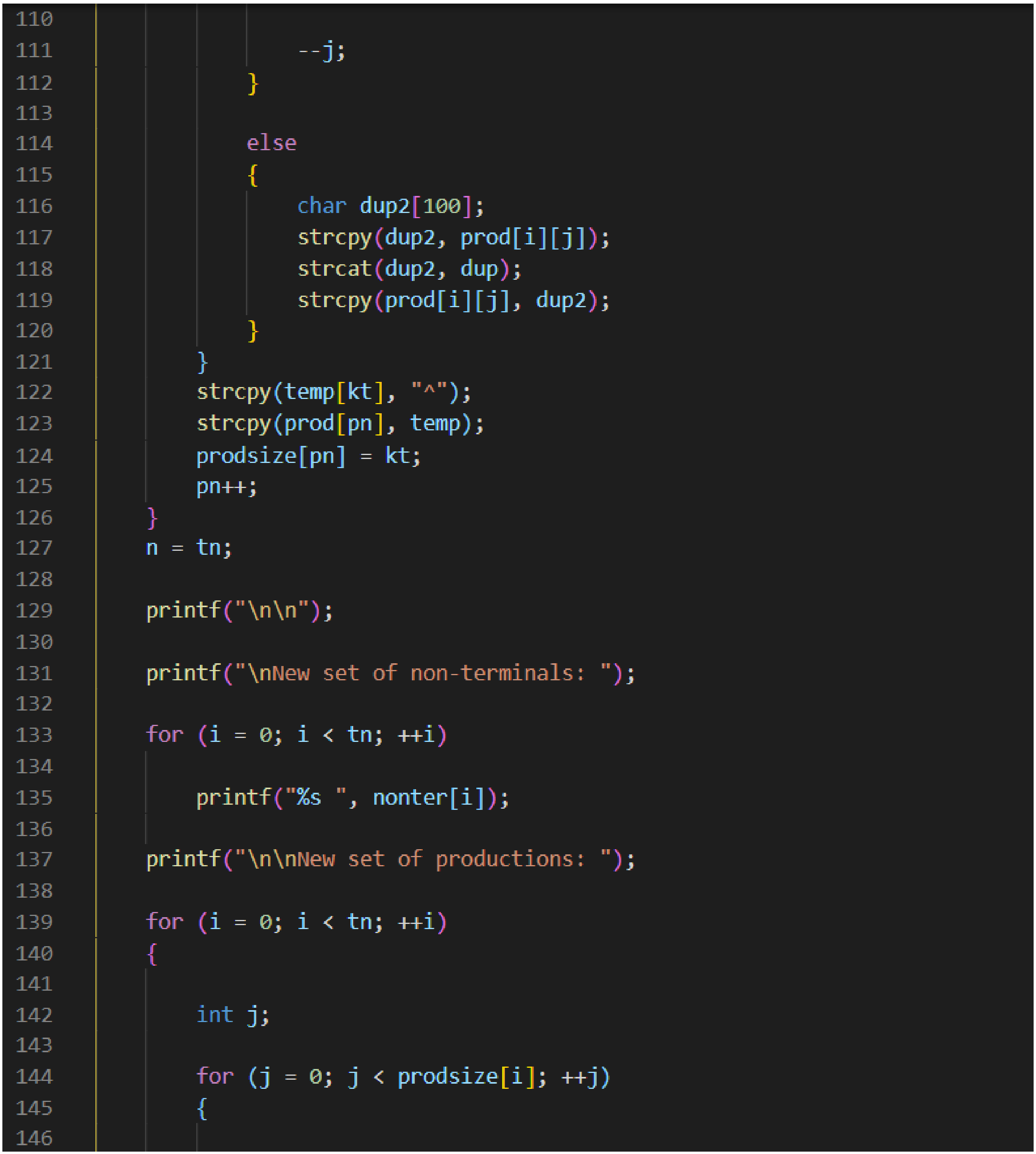
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char *substr(const char *src, int m, int n)
{
    int len = n - m;

    char *dest = (char *)malloc(sizeof(char) * (len + 1));

    for (int i = m; i < n && (*(src + i) != '\0'); i++)
    {
        *dest = *(src + i);
        dest++;
    }

    *dest = '\0';

    return dest - len;
}
int main()

{

    int n;
    printf("Prem Kumar\n191210037");
    printf("\nEnter number of non terminals: ");
    scanf("%d", &n);

    printf("\nEnter non terminals one by one: ");

    int i;

    char nonter[10][10];

    int leftrecr[10];
    for (int i = 0; i < 10; i++)
        leftrecr[i] = 0;
```



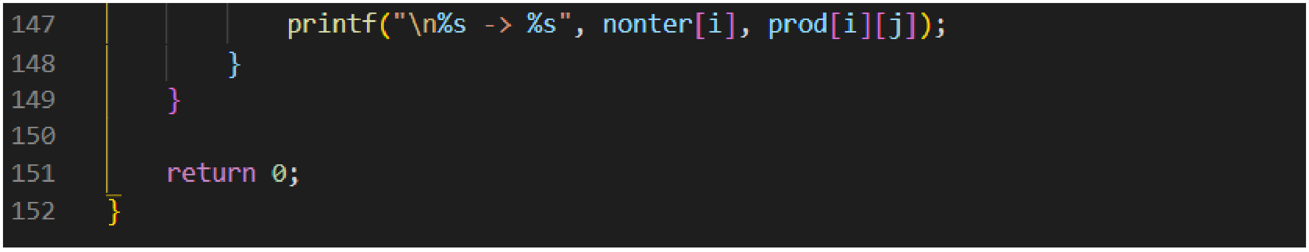
```c

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

            printf("\nNon terminal %d :", i + 1);

            scanf("%s", nonter[i]);
        }

        char prod[10][10][10];
        int prodsize[10];
        char dup[10];

        printf("\nEnter '^' for null/epsilon");

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

            printf("\nNumber of %s productions: ", nonter[i]);

            int k;

            scanf("%d", &k);
            prodsize[i] = k;
            int j;

            printf("\nOne by one enter all %s productions: ", nonter[i]);

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

                printf("\nRHS of production %d : ", j + 1);

                scanf("%s", prod[i][j]);

```



```
            if (strlen(nonter[i]) <= strlen(prod[i][j]) && strcmp(nonter[i], substr(prod[i][j], 0, strlen(nonter[i]))) == 0)

                leftrecr[i] = 1;
        }
    }

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

        printf("%d", leftrecr[i]);
    }
    int tn = n, pn = n;
    for (i = 0; i < n; ++i)
    {

        if (leftrecr[i] == 0)

            continue;

        int j;
        strcpy(dup, nonter[i]);
        strcat(dup, "'");
        strcpy(nonter[tn], dup);
        tn++;

        char temp[10][10];
        int kt = 0;
        for (j = 0; j < prodsize[i]; ++j)
        {

            if (strlen(nonter[i]) <= strlen(prod[i][j]) && strcmp(nonter[i], substr(prod[i][j], 0, strlen(nonter[i]))) == 0)
            {
                char dup2[100];
                strcpy(dup2, substr(prod[i][j], strlen(nonter[i]), strlen(prod[i][j]) - strlen(nonter[i])));
                strcpy(temp[kt], strcat(dup2, dup));
                kt++;
                strcpy(prod[i], memmove(&prod[i][j], &prod[i][j + 1], strlen(prod[i]) - j));
```

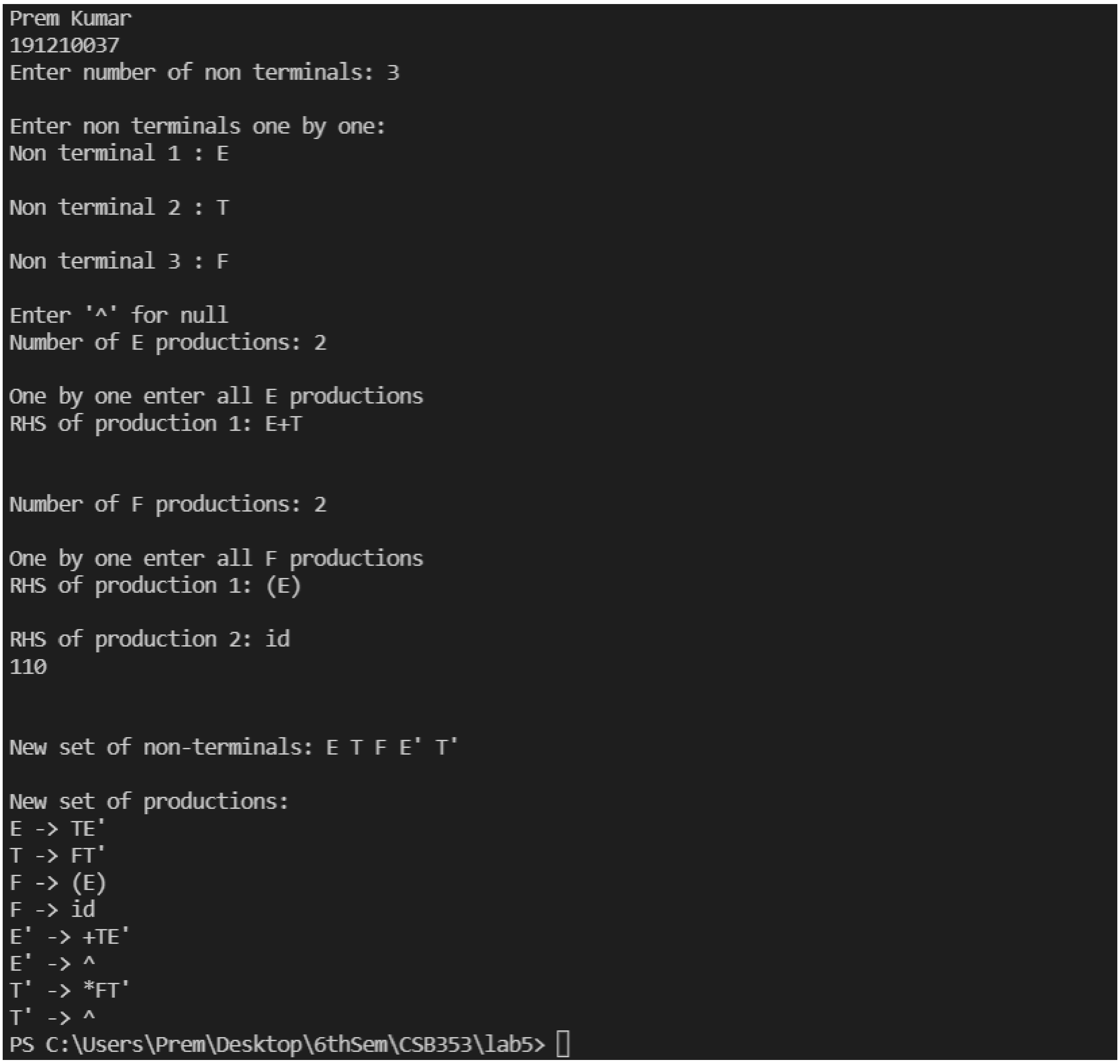


```c

                --j;
            }

            else
            {
                char dup2[100];
                strcpy(dup2, prod[i][j]);
                strcat(dup2, dup);
                strcpy(prod[i][j], dup2);
            }
        }
        strcpy(temp[kt], "^");
        strcpy(prod[pn], temp);
        prodsize[pn] = kt;
        pn++;
    }
    n = tn;

    printf("\n\n");

    printf("\nNew set of non-terminals: ");

    for (i = 0; i < tn; ++i)

        printf("%s ", nonter[i]);

    printf("\n\nNew set of productions: ");

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

        int j;

        for (j = 0; j < prodsize[i]; ++j)
        {

```



```c
            printf("\n%s -> %s", nonter[i], prod[i][j]);
        }
    }

    return 0;
}
```

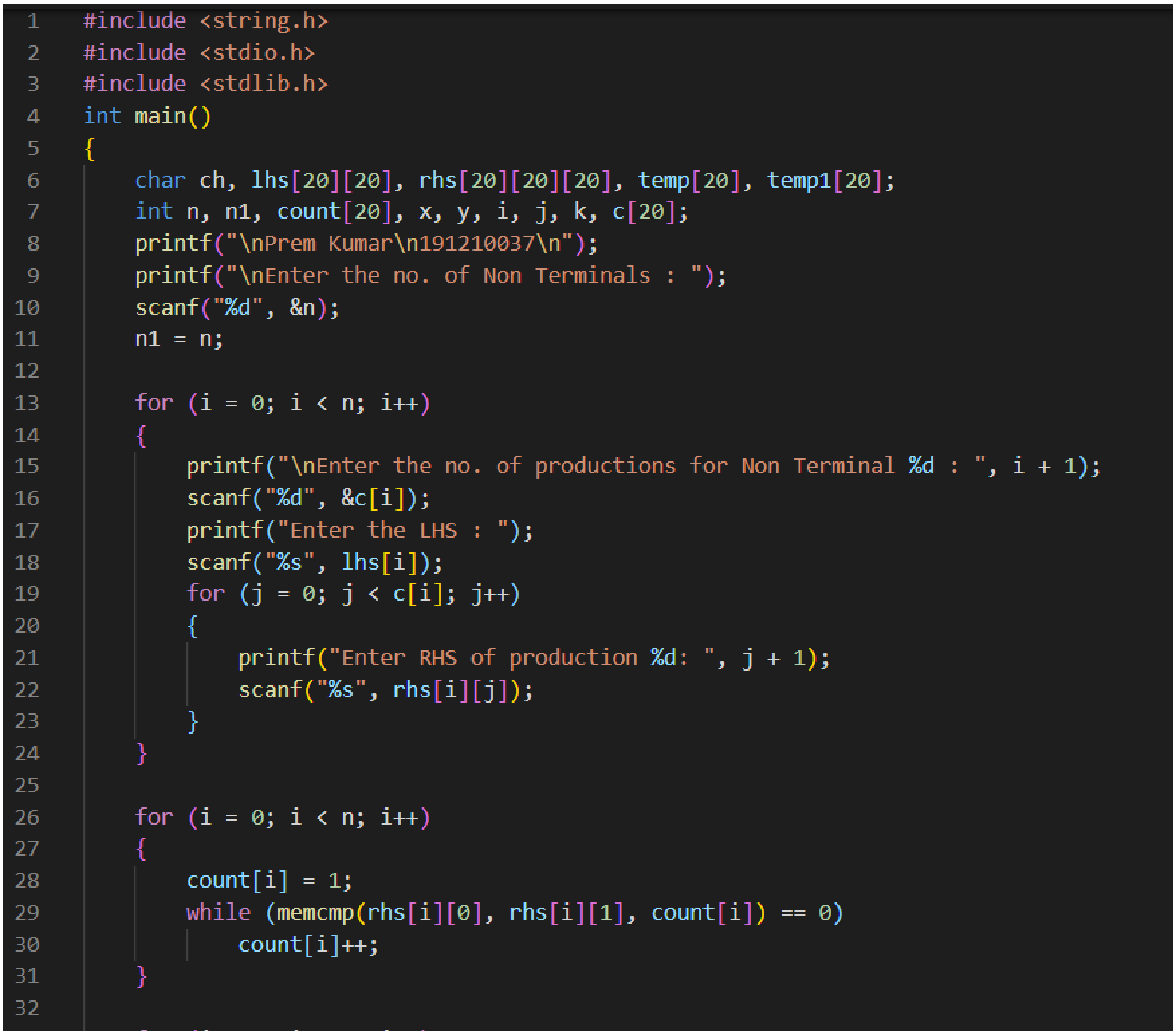
Output:



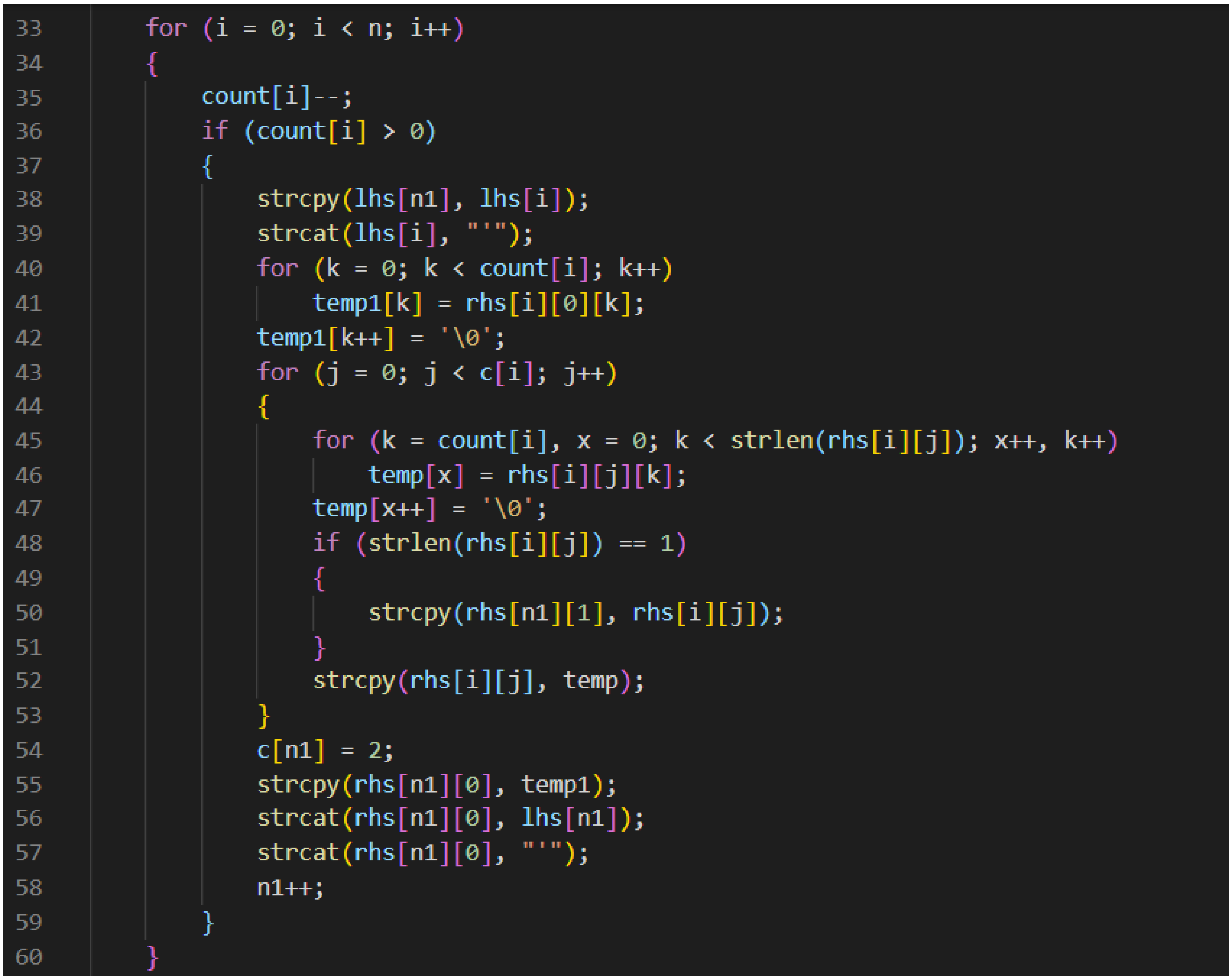
```
Prem Kumar
191210037
Enter number of non terminals: 3

Enter non terminals one by one:
Non terminal 1 : E

Non terminal 2 : T

Non terminal 3 : F

Enter '^' for null
Number of E productions: 2

One by one enter all E productions
RHS of production 1: E+T


Number of F productions: 2

One by one enter all F productions
RHS of production 1: (E)

RHS of production 2: id
110


New set of non-terminals: E T F E' T'

New set of productions:
E -> TE'
T -> FT'
F -> (E)
F -> id
E' -> +TE'
E' -> ^
T' -> *FT'
T' -> ^
PS C:\Users\Prem\Desktop\6thSem\CSB353\lab5> 
```

Ques 2. Write a C program that accepts a grammar and removes left factoring from it.

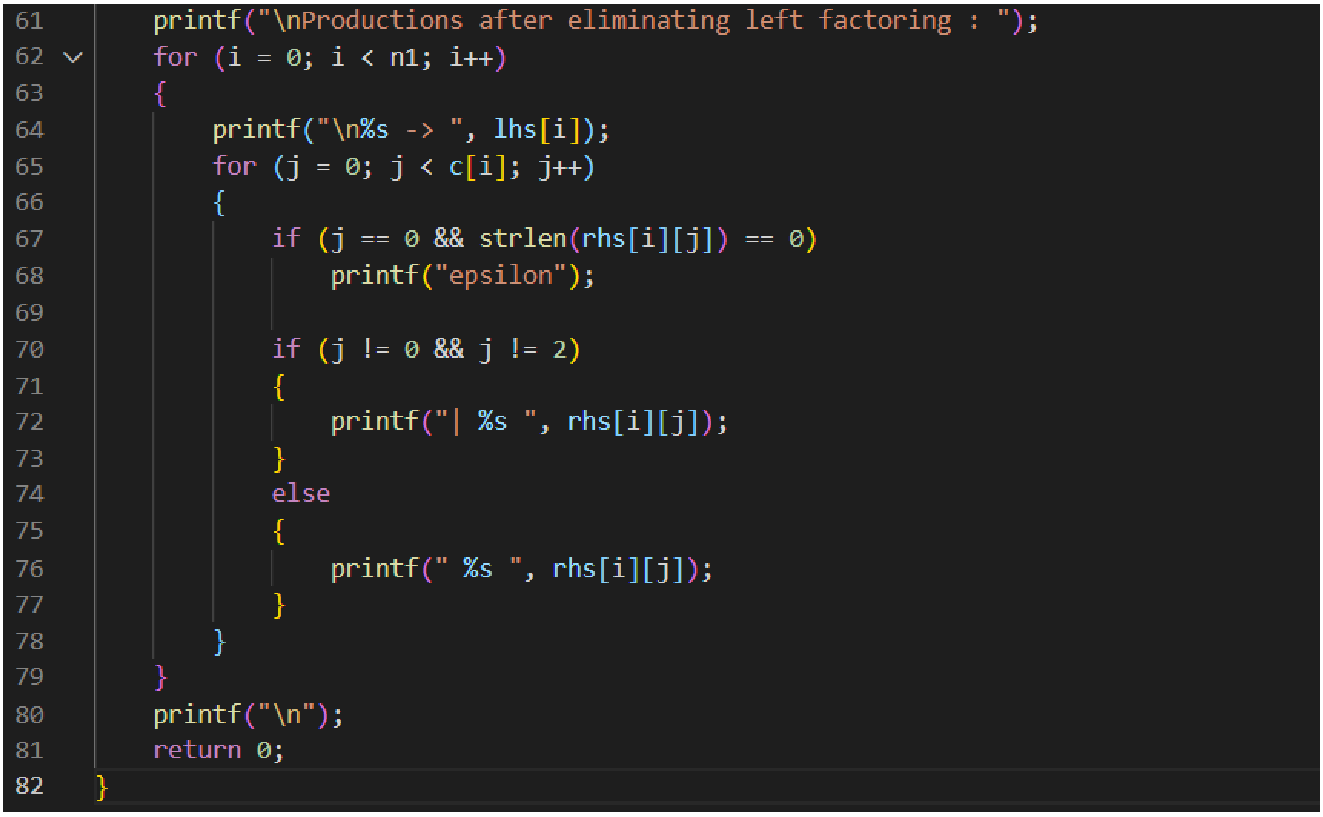
Code:



```c
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char ch, lhs[20][20], rhs[20][20][20], temp[20], temp1[20];
    int n, n1, count[20], x, y, i, j, k, c[20];
    printf("\nPrem Kumar\n191210037\n");
    printf("\nEnter the no. of Non Terminals : ");
    scanf("%d", &n);
    n1 = n;

    for (i = 0; i < n; i++)
    {
        printf("\nEnter the no. of productions for Non Terminal %d : ", i + 1);
        scanf("%d", &c[i]);
        printf("Enter the LHS : ");
        scanf("%s", lhs[i]);
        for (j = 0; j < c[i]; j++)
        {
            printf("Enter RHS of production %d: ", j + 1);
            scanf("%s", rhs[i][j]);
        }
    }

    for (i = 0; i < n; i++)
    {
        count[i] = 1;
        while (memcmp(rhs[i][0], rhs[i][1], count[i]) == 0)
            count[i]++;
    }

```

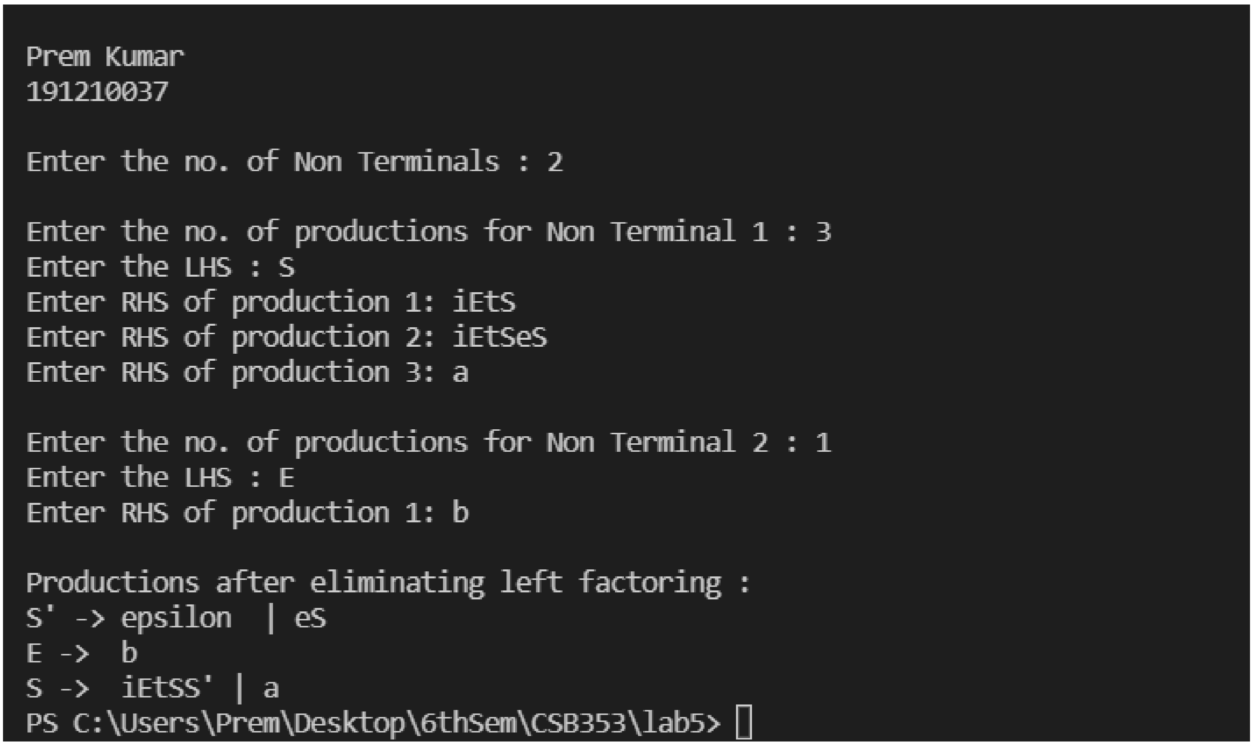


```c
    for (i = 0; i < n; i++)
    {
        count[i]--;
        if (count[i] > 0)
        {
            strcpy(lhs[n1], lhs[i]);
            strcat(lhs[i], "'");
            for (k = 0; k < count[i]; k++)
                temp1[k] = rhs[i][0][k];
            temp1[k++] = '\0';
            for (j = 0; j < c[i]; j++)
            {
                for (k = count[i], x = 0; k < strlen(rhs[i][j]); x++, k++)
                    temp[x] = rhs[i][j][k];
                temp[x++] = '\0';
                if (strlen(rhs[i][j]) == 1)
                {
                    strcpy(rhs[n1][1], rhs[i][j]);
                }
                strcpy(rhs[i][j], temp);
            }
            c[n1] = 2;
            strcpy(rhs[n1][0], temp1);
            strcat(rhs[n1][0], lhs[n1]);
            strcat(rhs[n1][0], "'");
            n1++;
        }
    }
```



```c
    printf("\nProductions after eliminating left factoring : ");
    for (i = 0; i < n1; i++)
    {
        printf("\n%s -> ", lhs[i]);
        for (j = 0; j < c[i]; j++)
        {
            if (j == 0 && strlen(rhs[i][j]) == 0)
                printf("epsilon");

            if (j != 0 && j != 2)
            {
                printf("| %s ", rhs[i][j]);
            }
            else
            {
                printf(" %s ", rhs[i][j]);
            }
        }
    }
    printf("\n");
    return 0;
}
```

Output:



```
Prem Kumar
191210037

Enter the no. of Non Terminals : 2

Enter the no. of productions for Non Terminal 1 : 3
Enter the LHS : S
Enter RHS of production 1: iEtS
Enter RHS of production 2: iEtSeS
Enter RHS of production 3: a

Enter the no. of productions for Non Terminal 2 : 1
Enter the LHS : E
Enter RHS of production 1: b

Productions after eliminating left factoring :
S' -> epsilon  | eS
E ->  b
S ->  iEtSS' | a
PS C:\Users\Prem\Desktop\6thSem\CSB353\lab5>
```