Dept. of Computer Science & Engineering

আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম

الجامعة الإسلامية العالمية شيتاغونغ

International Islamic University Chittagong

Course Code:CSE-3528

Course Title: Compiler Lab

Spring-22

Submitted By:

Name: Farida Nusrat

ID: C201242

Semester: 5th

Section: 5AF

Email: c201242@ugrad.iiuc.ac.bd

Submitted To:

Mrs. Israt Binteh Habib

Dept. Of CSE,IIUC.

Signature: Farida Nusrat

Submission Date:17.10.22

OBJECTIVE:

Implement left recursion elimination from a grammar.

RESOURCE:

Code-blocks

PROGRAM LOGIC:

1. Firstly we will include required directives.
2. Then we will declare the required variables.
3. Then we will enter the parent non-terminal.
4. After that we will enter the number of productions and then we will enter the productions.
5. We will check the production have left recursion or not.
6. If we have the left-recursive pair of productions, then we can eliminate left recursion by replacing the pair of productions.
7. Then we have converted the grammar into a right recursive grammar and we have done the elimination of left recursion.

PROCEDURE:

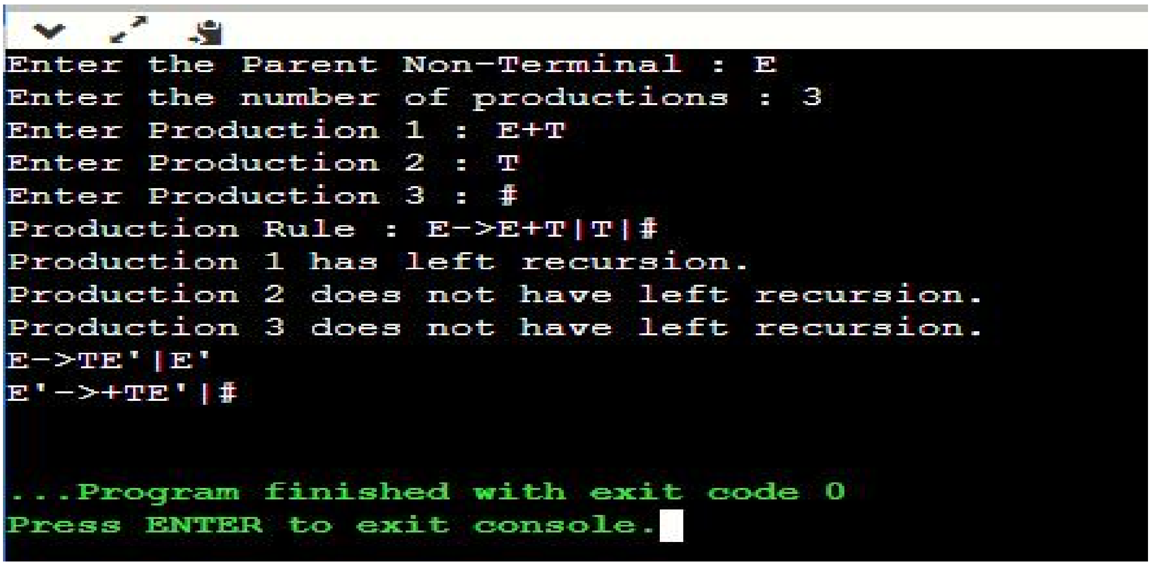
Go to debug -> run or press CTRL + F9 to run the program.

PROGRAM:

```
#include <bits/stdc++.h>

using namespace std;

int main()

{

    string ip,op1,op2,temp;

    int sizes[10] = {};

    char c;

    int n,j,l;

    cout<<"Enter the Parent Non-Terminal : ";

    cin>>c;

    ip.push_back(c);

    op1 += ip + "\'->";

    ip += "->";

    op2+=ip;

    cout<<"Enter the number of productions : ";
```

```
cin>>n;
for(int i=0; i<n; i++)
{   cout<<"Enter Production "<<i+1<<" : ";
    cin>>temp;
    sizes[i] = temp.size();
    ip+=temp;
    if(i!=n-1)
        ip += "|";
}
cout<<"Production Rule : "<<ip<<endl;
for(int i=0,k=3; i<n; i++)
{
    if(ip[0] == ip[k])
    {
        cout<<"Production "<<i+1<<" has left recursion."<<endl;
        if(ip[k] != '#')
        {
            for(l=k+1; l<k+sizes[i]; l++)
                op1.push_back(ip[l]);
            k=l+1;
            op1.push_back(ip[0]);
            op1 += "\'|";
        }
    }
    else
    {
        cout<<"Production "<<i+1<<" does not have left recursion."<<endl;
        if(ip[k] != '#')
        {
          for(j=k; j<k+sizes[i]; j++)

            op2.push_back(ip[j]);

          k=j+1;

          op2.push_back(ip[0]);

          op2 += "\'|";

        }

        else

        {

          op2.push_back(ip[0]);

          op2 += "\'";

        }

  }

  }

    op1 += "#";

    cout<<op2<<endl;

    cout<<op1<<endl;

    return 0;

  }
```

INPUT & OUTPUT:



```
Enter the Parent Non-Terminal : E
Enter the number of productions : 3
Enter Production 1 : E+T
Enter Production 2 : T
Enter Production 3 : #
Production Rule : E->E+T|T|#
Production 1 has left recursion.
Production 2 does not have left recursion.
Production 3 does not have left recursion.
E->TE'|E'
E'->+TE'|#


...Program finished with exit code 0
Press ENTER to exit console.
```

OUTCOMES:

The implementation is done successfully.