AIM :- WAP for Left recursion problem Removal

PRODEDURE:-

- Write algorithm to remove left recursion problem
- Generate a program to remove left recursion problem from any given grammar rules

Program :-

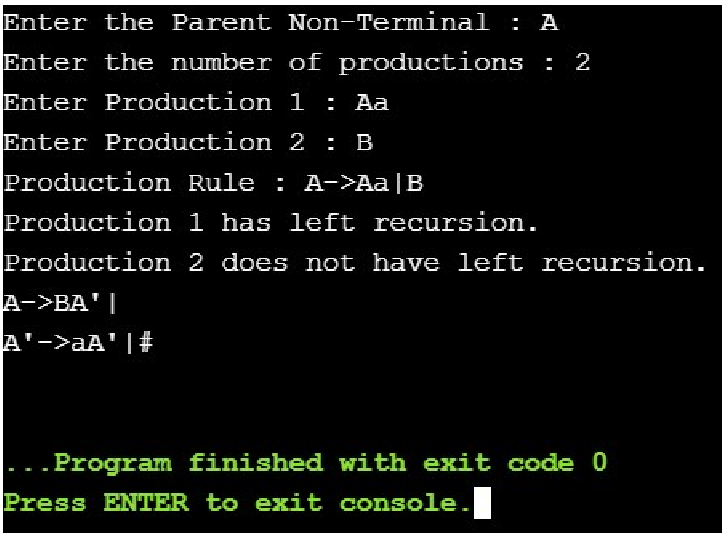
left_recursion.cpp

```
#include<iostream>

#include<string>

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;
```

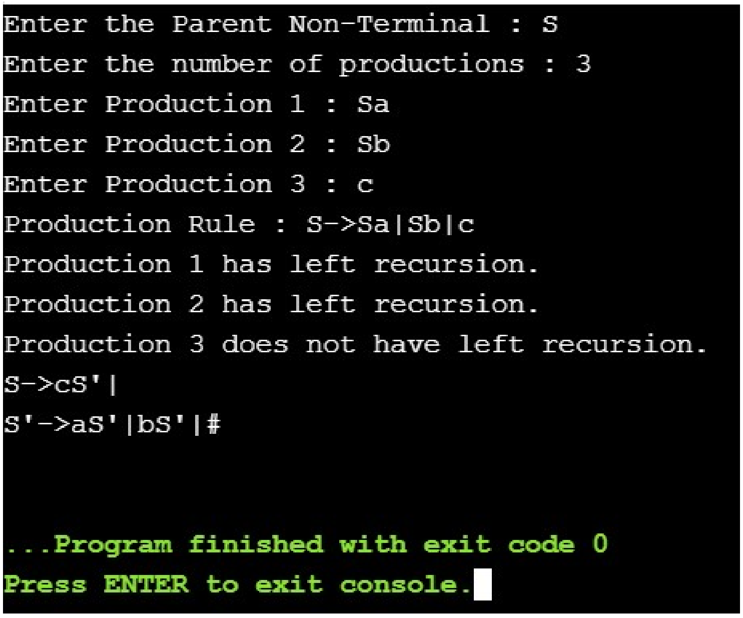
```
for(int i=0,k=3;i<n;i++)
{
   if(ip[0] == ip[k])
   {
      cout<<"Production "<<i+1<<" has left recursion."<<endl;
      if(ip[k] != '#')
      {
         for(l=k+1;l<k+sizes[i];l++)
            op1.push_back(ip[l]);
         k=l+1;
         op1.push_back(ip[0]);
         op1 += "\'|";
      }
   }
   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;}
```

Output:



```
Enter the Parent Non-Terminal : A
Enter the number of productions : 2
Enter Production 1 : Aa
Enter Production 2 : B
Production Rule : A->Aa|B
Production 1 has left recursion.
Production 2 does not have left recursion.
A->BA'|
A'->aA'|#


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



```
Enter the Parent Non-Terminal : S
Enter the number of productions : 3
Enter Production 1 : Sa
Enter Production 2 : Sb
Enter Production 3 : c
Production Rule : S->Sa|Sb|c
Production 1 has left recursion.
Production 2 has left recursion.
Production 3 does not have left recursion.
S->cS'|
S'->aS'|bS'|#


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