7a: Left Recursion for expression grammar using C++.

/\*

E->E+T/T

T->T\*F/F

F->(E)/id

\*/

#include <bits/stdc++.h>

using namespace std;

void removeLeftRecursion(string str) {

int ind = 0;

string s1="",s2="";

for (int i = 0; i < str.size(); i++) {

if (str[i] == '/') {

ind = i;

break;

}

}

if(!isupper(str[0]))

{

cout<<"Invalid Production";

}

else if (str[0] == str[3] && ind != 0) {

s1 = string(1,str[0]) + " -> " + string(1, str[ind + 1]) + string(1, str[0]) + "'";

s2 = string(1, str[0]) + "'"+ " -> "+str.substr(4,str.size()-ind) + string(1, str[0]) + "'"+string(1, str[ind])+"ε";

cout << s1<<"\n";

cout<< s2 <<"\n";

}

else

{

cout<<str<<"\n";

}

}

int main() {

int n;

cout<<"Enter your number of production: ";

cin>>n;

string str[n];

cout << "Enter a grammar: \n";

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

{

cin >> str[i];

}

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

{

removeLeftRecursion(str[i]);

}

}