Left recursion in c:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define SIZE 20

int main()

{

char pro[SIZE], alpha[SIZE], beta[SIZE];

int nont\_terminal,i,j, index=3;

printf("Enter the Production as E->E|A: ");

scanf("%s", pro);

nont\_terminal=pro[0];

if(nont\_terminal==pro[index])

{

for(i=++index,j=0;pro[i]!='|';i++,j++){

alpha[j]=pro[i];

if(pro[i+1]==0){

printf("This Grammar CAN'T BE REDUCED.\n");

exit(0);

}

}

alpha[j]='\0';

if(pro[++i]!=0)

{

for(j=i,i=0;pro[j]!='\0';i++,j++){

beta[i]=pro[j];

}

beta[i]='\0';

printf("\nGrammar Without Left Recursion: \n\n");

printf(" %c->%s%c'\n", nont\_terminal,beta,nont\_terminal);

printf(" %c'->%s%c'|#\n", nont\_terminal,alpha,nont\_terminal);

}

else

printf("This Grammar CAN'T be REDUCED.\n");

}

else

printf("\n This Grammar is not LEFT RECURSIVE.\n");

}

Output:

