//Expt 4

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class LL1

{

public static String prod[] = new String[10];

public static char fst[][] = new char[10][4];

public static char stack[] = new char[10];

public static int n,i,top=0,j,k,l,t;

public static char c,temp,next;

public static String ipstr,str;

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("\nEnter the no. of productions: ");

n = sc.nextInt();

System.out.println("\nEnter the productions:");

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

prod[i]=sc.next();

calFirst();

calFollow();

System.out.println("\nFirst-Follow Table:");

System.out.println("\n\tNonT \tFirst \tFollow");

for( i=top-1; i>=0; i-- )

System.out.println("\t"+fst[i][0]+"\t"+fst[i][1]+"\t"+fst[i][2]);

System.out.print("\nEnter input string: ");

ipstr = sc.next();

LL();

}

public static void calFirst()

{

for(i=n-1;i>=0;i--)

{

c = prod[i].charAt(2);

if(c>='A' && c<='Z')

{

temp = find( c, 1 );

fst[top][0] = prod[i].charAt(0);

fst[top][1] = temp;

}

else

{

fst[top][0] = prod[i].charAt(0);

fst[top][1] = c;

}

top++;

}

}

public static void calFollow()

{

int check=1,index;

fst[top-1][2] = '$';

do

{

check = 1;

for( j=0; j<top; j++ )

{

if( fst[j][2] == '\0')

{

c = fst[j][0];

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

{

index = prod[i].indexOf(c);

if( index > 0 )

{

if( prod[i].length() == index+1)

temp = find( prod[i].charAt(0) , 2 );

else

{

next = prod[i].charAt(index+1);

if( next>='A' && next<='Z')

temp = find( next , 2 );

else

temp = next;

}

fst[j][2] = temp;

}

}

}

}

for(j=0;j<top;j++)

if(fst[j][2]=='\0')

check=0;

}while(check==0);

}

public static void LL()

{

char ip[] = new char[10];

 for( i=0; i<ipstr.length(); i++ )

ip[i] = ipstr.charAt(i);

ip[i] = '$';

l=0;

t=1;

stack[0] = '$';

stack[1] = prod[0].charAt(0);

System.out.println("\nThe Parsing Table:");

System.out.println("\n\tStack \t\tInput ");

System.out.print("\t");

while( stack[t]!='$' )

{

for( i=t; i>=0; i-- )

System.out.print(stack[i]);

System.out.print("\t\t");

for( i=l; i<=ipstr.length(); i++ )

System.out.print(ip[i]);

System.out.println();

System.out.print("\t");

if( stack[t]>='A' && stack[t]<='Z' )

{

c = find(stack[t],1);

if( c != ip[l] )

break;

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

if( prod[i].charAt(0) == stack[t] )

{

t--;

for( j=prod[i].length()-1; j>=2; j-- )

{

t++;

stack[t] = prod[i].charAt(j);

}

}

}

else if( ip[l]==stack[t] )

{

l++;

t--;

}

else

{

break;

}

}

for( i=t; i>=0; i-- )

System.out.print(stack[i]);

System.out.print("\t\t");

for( i=l; i<=ipstr.length(); i++ )

System.out.print(ip[i]);

System.out.println();

if(stack[t]=='$' && ip[l]=='$')

System.out.println("String is accepted by the Grammar");

else

System.out.println("\nString is rejected");

}

public static char find(char ch, int z)

{

for( k=0; k<top; k++ )

if( ch==fst[k][0] )

return fst[k][z];

return '\0';

}

}

/\*

OUTPUT

Enter the no. of productions: 3

Enter the productions:

S-aAa

A-bBb

B-c

First-Follow Table:

NonT First Follow

S a $

A b a

B c b

Enter input string: abcba

The Parsing Table:

Stack Input

S$ abcba$

aAa$ abcba$

Aa$ bcba$

bBba$ bcba$

Bba$ cba$

cba$ cba$

ba$ ba$

a$ a$

$ $

String is accepted by the Grammar

\*/