#include<bits/stdc++.h>

using namespace std;

int solve(string s1, string s2){

int n = s1.size();

vector<vector<int> > dpt(n,vector<int>(n,0));

vector<vector<int> > dpf(n,vector<int>(n,0));

for(int g=0;g<n;g++){

for(int i=0,j=g;j<n; i++,j++){

if(g==0){

char ch = s1[i];

if(ch=='T'){

dpt[i][j] = 1;

dpf[i][j] = 0;

}

else{

dpt[i][j] = 0;

dpf[i][j] = 1;

}

}

else{

for(int k=i;k<j;k++){

char oprt = s2[k];

int ltc = dpt[i][k];

int rtc = dpt[k+1][j];

int lfc = dpf[i][k];

int rfc = dpf[k+1][j];

if(oprt=='&'){

dpt[i][j] += (ltc\*rtc);

dpf[i][j] += (ltc\*rfc + lfc\*rtc + lfc\*rfc);

}

else if(oprt=='|'){

dpt[i][j] += (ltc\*rfc + ltc\*rtc + lfc\*rtc);

dpf[i][j] += (lfc\*rfc);

}

else{// xor ^

dpt[i][j] += (ltc\*rfc + lfc\*rtc);

dpf[i][j] += (lfc\*rfc + ltc\*rtc);

}

}

}

}

}

return dpt[0][n-1];

}

int main(){

string s1,s2;

cin>>s1>>s2;

cout<<solve(s1,s2);

}