**Simpsons’ Hidden Talents**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 5697    Accepted Submission(s): 2057**

Problem Description

Homer: Marge, I just figured out a way to discover some of the talents we weren’t aware we had.  
Marge: Yeah, what is it?  
Homer: Take me for example. I want to find out if I have a talent in politics, OK?  
Marge: OK.  
Homer: So I take some politician’s name, say Clinton, and try to find the length of the longest prefix  
in Clinton’s name that is a suffix in my name. That’s how close I am to being a politician like Clinton  
Marge: Why on earth choose the longest prefix that is a suffix???  
Homer: Well, our talents are deeply hidden within ourselves, Marge.  
Marge: So how close are you?  
Homer: 0!  
Marge: I’m not surprised.  
Homer: But you know, you must have some real math talent hidden deep in you.  
Marge: How come?  
Homer: Riemann and Marjorie gives 3!!!  
Marge: Who the heck is Riemann?  
Homer: Never mind.  
Write a program that, when given strings s1 and s2, finds the longest prefix of s1 that is a suffix of s2.

Input

Input consists of two lines. The first line contains s1 and the second line contains s2. You may assume all letters are in lowercase.

Output

Output consists of a single line that contains the longest string that is a prefix of s1 and a suffix of s2, followed by the length of that prefix. If the longest such string is the empty string, then the output should be 0.  
The lengths of s1 and s2 will be at most 50000.

Sample Input

clinton

homer

riemann

marjorie

Sample Output

0

rie 3

分析：  
给两个字符串，s1和s2 求s1的最长前缀等于s2的后缀

此题还是next数组的应用 next[j]数组存的就是前j个字符前缀与后缀的最长的匹配长度 所以把两个字符串链接起来 再求next数组 就可以了

但是也有可能是你求出的答案会大于最短字符串的长度；

列如：

abcabcabcabc  
abcabcabcabcabc

答案就是 12；

所以在求出答案后 还要与字符的长度进行比较（最长的长度也不会超过两者之间最短的那个字符串）

AC代码：

#include <stdio.h>

#include <string.h>

#define MOD 10007

char a[200005],b[200005],c[200005];

int next[200005];

int n,m,sum;

void Next() ///求next数组

{

next[0] = next[1] = 0;

for(int i = 1; i < n+m; i++)

{

int j = next[i];

while(j&&c[j]!=c[i])

j = next[j];

next[i+1] = c[i]==c[j]?j+1:0;

}

if(next[n+m]==0)

printf("0\n");

else

{

int flag = next[n+m];

a[flag] = '\0';

if(n > m)

{

if(flag > m)

printf("%s %d\n",b,m);

else

printf("%s %d\n",a,flag);

}

else

{

if(flag > n)

printf("%s %d\n",a,n);

else

printf("%s %d\n",a,flag);

}

}

}

int main()

{

while(~scanf("%s" ,a))

{

scanf("%s" ,b);

n = strlen(a);

m = strlen(b);

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

c[i] = a[i];

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

c[n+i] = b[i];

Next();

}

return 0;

}