题目描述

妮妮面前有一个巨大的宝箱，宝箱上有一个锁，锁上有一串仅包含小写字母的字符串 s 。妮妮在寻宝的过程中获得了 m 个不同的钥匙，每个钥匙上也有一串仅包含小写字母的字符串 ti 。
由于锁的独特构造，妮妮可以使用任意一把钥匙打开宝箱，但是从宝箱中获得的金币数量并不相同。使用第 i 个钥匙会获得 wi 个金币，其中 wi 为 ti 在 s 中出现的次数。
由于宝箱只能开一次，妮妮想知道打开一次宝箱最多可以获得多少金币。

输入格式

第一行一个字符串 s（1≤|s|≤105），表示锁上的字符串。
第二行一个正整数 m（1≤m≤100），表示钥匙的个数。
接下来 m 行每行一个字符串 ti（1≤|ti|≤105），表示第 i 个钥匙上的字符串。

输出格式

输出一行一个整数，表示妮妮打开一次宝箱最多获得的金币个数

解题思路

KMP算法匹配m次

AC代码

```
/*
 Author: 尹立健
 Result: AC Submission_id: 5933338
 Created at: Wed Nov 29 2023 20:30:41 GMT+0800 (China Standard Time)
 Problem_id: 7509   Time: 453   Memory: 4508
*/

#include<bits/stdc++.h>
using namespace std;
vector<int> prefix_function(string s) {
  int n = (int)s.length();
  vector<int> pi(n);
  for (int i = 1; i < n; i++) {
    int j = pi[i - 1];
    while (j > 0 && s[i] != s[j]) j = pi[j - 1];
    if (s[i] == s[j]) j++;
    pi[i] = j;
  }
  return pi;
}
int find_occurrences(string text, string pattern) {
  string cur = pattern + '#' + text;
  int times=0;
  int sz1 = text.size(), sz2 = pattern.size();
  vector<int> v;
```

```
  vector<int> lps = prefix_function(cur);
  for (int i = sz2 + 1; i <= sz1 + sz2; i++) {
    if (lps[i] == sz2) {
        times++;
    }
  }
  return times;
}
int main()
{
    int t,maxN=0;
    char S[105000];
    char T[105000];
    scanf("%s",S);
    scanf("%d",&t);
    while(t--){
        memset(T,0,sizeof(T));
        scanf("%s",T);
        maxN=max(maxN,find_occurrences(S,T));
    }
    printf("%d",maxN);
    return 0;
}
```