

妮妮与宝箱

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 t_i 。

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 w_i 个金币，其中 w_i 为 t_i 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 ($1 \leq |s| \leq 10^5$)，表示锁上的字符串。

第二行一个正整数 m ($1 \leq m \leq 100$)，表示钥匙的个数。

接下来 m 行每行一个字符串 t_i ($1 \leq |t_i| \leq 10^5$)，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。

题解

对于每个 t_i ，进行一次 KMP 匹配并记录出现次数。时间复杂度 $O(m|s| + m|t|)$ ， $|s|$ 和 $|t_i|$ 规模同阶，可以认为复杂度为 $O(nm)$ ， n 是字符串长度。空间复杂度 $O(n)$


```

int read()
{
    char c = getchar();
    int w = 0, f = 0;
    while (!isdigit(c))
    {
        if (c == '-')
            f = 1;
        c = getchar();
    }
    while (isdigit(c))
    {
        w = w * 10 + c - '0';
        c = getchar();
    }
    return f ? -w : w;
}

#define N 5000005
int pi[N];
char s[N], t[N], p[N];
int n, m;
int T, ans;
int main()
{
    scanf("%s", s);
    T = read();
    while (T--)
    {
        scanf("%s", t);
        n = strlen(s), m = strlen(t);
        for (int i = 0; i < m; i++)
            p[i] = t[i];
        p[m] = '?';
        for (int i = 0; i < n; i++)
        {
            p[i + m + 1] = s[i];
        }
        int len = n + m + 1;
        for (int i = 0; i < len; i++)
            pi[i] = 0;
        for (int i = 1; i < len; i++)
        {
            int j = pi[i - 1];
            while (j > 0 && p[i] != p[j])
            {
                j = pi[j - 1];
            }
            if (p[i] == p[j])
                j++;
            pi[i] = j;
        }
        int cnt = 0;
        for (int i = m + 1; i < len; i++)
        {
            if (pi[i] == m)
            {
                cnt++;
            }
        }
        if (cnt > ans)
            ans = cnt;
    }
    printf("%d\n", ans);
    return 0;
}

```

```
}

```