

**Notice:** 1:由于本OJ建立在Linux平台下，而许多题的数据在Windows下制作，请注意输入、输出语句及数据类型及范围，避免无谓的RE出现。2:本站即将推出针对初学者的试题系统(与目前OJ是分开的，互不影响)，内容覆盖从语法入门到NOI的所有知识点，敬请关注。

## 3099: Hash Killer III

Time Limit: 10 Sec Memory Limit: 128 MBSec Special Judge

Submit: 562 Solved: 0

[\[Submit\]](#)[\[Status\]](#)[\[Discuss\]](#)

### Description

这天天气不错，hzhwcmhf神犇给VFleaKing出了一道题：  
给你一个长度为N的字符串S，求有多少个不同的长度为L的子串。  
子串的定义是S[i]、S[i + 1]、... S[r]这样连续的一段。  
两个字符串被认为是不同的当且仅当某个位置上的字符不同。

VFleaKing一看觉得这不是Hash的裸题么！于是果断写了哈希 + 排序。  
而hzhwcmhf神犇心里自然知道，这题就是后缀数组的height中  $< L$  的个数 + 1，就是后缀自动机上代表的长度区间包含L的结点个数，就是后缀树深度为L的结点的数量。  
但是hzhwcmhf神犇看了看VFleaKing的做法表示非常汗。于是想卡掉他。

VFleaKing使用的是字典序哈希，其代码大致如下：

```
u64 val = 0;
for (int i = 0; i < l; i++)
    val = (val * base + s[i] - 'a') % Mod;
u64是无符号int64，范围是 $[0, 2^{64})$ 。
```

base是一个常量，VFleaKing会根据心情决定其值。

Mod是int32范围内的素数。即Mod范围是 $[0, 2^{31})$ ，且是个素数。

VFleaKing还求出来了 $\text{base}^l \% \text{Mod}$ ，即base的l次方除以Mod的余数，这样就能方便地求出所有长度为L的子串的哈希值。

然后VFleaKing给哈希值排序，去重，求出有多少个不同的哈希值，把这个数作为结果。

但是VFleaKing意识到这样会被坑，于是采用了两个Mod来解决问题。两个Mod模出来的值都相同才被认为是相同。

其算法的C++代码如下：

```

typedef unsigned long long u64;
typedef pair<int, int> PII;

const int MaxN = 100000;

inline int hash_handle(
    const char *s, const int &n, const int &l, const int &base,
    const int &mod1, const int &mod2)
{
    int li_n;
    static PII li[MaxN];
    u64 hash_pow_l;
    u64 val;

    hash_pow_l = 1;
    for (int i = 1; i <= l; i++)
        hash_pow_l = (hash_pow_l * base) % mod1;

    li_n = 0;
    val = 0;
    for (int i = 0; i < l; i++)
        val = (val * base + s[i] - 'a') % mod1;
    li[li_n++].first = val;
    for (int i = l; i < n; i++)
    {
        val = (val * base + s[i] - 'a') % mod1;
        val = (val + mod1 - ((s[i] - l) - 'a') * hash_pow_l) % mod1) % mod1;
        li[li_n++].first = val;
    }

    hash_pow_l = 1;
    for (int i = 1; i <= l; i++)
        hash_pow_l = (hash_pow_l * base) % mod2;

    li_n = 0;
    val = 0;
    for (int i = 0; i < l; i++)
        val = (val * base + s[i] - 'a') % mod2;
    li[li_n++].second = val;
    for (int i = l; i < n; i++)
    {
        val = (val * base + s[i] - 'a') % mod2;
        val = (val + mod2 - ((s[i] - l) - 'a') * hash_pow_l) % mod2) % mod2;
        li[li_n++].second = val;
    }
}

```

```
}  
  
sort(li, li + li_n);  
li_n = unique(li, li + li_n) - li;  
return li_n;  
}
```

hzhwcmhf当然知道怎么卡啦！但是他想考考你。

## Input

没有输入。

## Output

你需要输出一组数据使得VFleaKing的代码WA掉。我们会使用Special Judge检查你的结果的正确性。

输出文件共两行。

第一行两个用空格隔开的数 $n$ 、 $l$ 。

第二行是一个长度为 $n$ 的字符串。只能包含'a'~'z'。

需要保证 $1 \leq n \leq 10^5$ ,  $1 \leq l \leq n$ ,

不符合以上格式会WA。

不要有多余字符，很可能导致你WA。

## Sample Input

没有

## Sample Output

8 4

buaabuaa

(当然这个输出是会WA的)

## HINT

其实出题人并不知道怎么做.....只是把这题挂在这里希望有人能给个解答。

如果这个问题解决了hash应该能算是正式被hack了。

## Source

VFleaKing & hzhwcmhf

[\[Submit\]](#)[\[Status\]](#)[\[Discuss\]](#)

[HOME](#) [Back](#)

---

[한국어](#) [中文](#) [فارسی](#) [English](#) [ไทย](#)

版权所有 ©2008-2012 大视野在线测评 | 湘ICP备13009380号 | 站长统计

Based on opensource project hustoj.