

Day1

题目名称	古代密码	塔	数球
文件名	cryptogram	tower	ball
内存限制	512MB	512MB	512MB
时间限制	1s	1s	1s

注：

- (1) 没有特殊说明，所有输入的整数均为 int 范围内的非负整数。
- (2) 输入文件均为文件名.in/.out。

古代密码

【题目描述】

古罗马帝国有一个拥有各种部门的强大政府组织。其中一个部门就是保密服务部门。为了保险起见，在省与省之间传递的重要文件中的大写字母是加密的。当时最流行的加密方法是替换和重新排列。

替换方法是将所有出现的字符按照一个规则替换，比如 ABCDEFGHIJKLMNOPQRSTUVWXYZ 到 BCDEFGHIJKLMNOPQRSTUVWXYZA，如果原词是 "VICTORIOUS" 则它变成 "WJDUPSJPVT"。

排列方法改变原来单词中字母的顺序。例如：将顺序<2, 1, 5, 4, 3, 7, 6, 10, 9, 8>应用到 "VICTORIOUS" 上，则得到"IVOTCIRSUO"。

人们很快意识到单独应用替换方法或排列方法加密，都是很不保险的。但是如果结合这两种方法，在当时就可以得到非常可靠的加密方法。所以，很多重要信息先使用替换方法加密，再将加密的结果用排列的方法加密。用两种方法结合就可以将"VICTORIOUS" 加密成 "JWPUDJSTVP"。

考古学家最近在一个石台上发现了一些信息。初看起来它们毫无意义，所以有人设想它们可能是用替换和排列的方法被加密了。人们试着解读了石台上的密码，现在他们想检查解读的是否正确。他们需要一个计算机程序来验证，你的任务就是写这个验证程序。

【输入描述】

输入有两行。第一行是石台上的文字。文字中没有空格，并且只有大写英文字母。第二行是被解读出来的加密前的文字。第二行也是

由大写英文字母构成的。

两行字符数目的长度都不超过 100。

【输出描述】

如果第二行经过某种加密方法后可以产生第一行的信息，输出 "YES"，否则输出 "NO"。

【样例】

cryptogram. in	cryptogram. out
JWPUDJSTVP	YES
VICTORIOUS	

【数据范围】

对于 30%的数据：字符串长度 ≤ 10

对于 50%的数据：字符串长度 ≤ 50

对于 100%的数据：字符串长度 ≤ 100

塔

【题目描述】

小 A 想搭一个体积不超过 m 的塔，他有各种大小的立方积木，比如边长为 a 的积木，体积为 a^3 ，现在小 A 需要你给一个 X ，每次小 A 会用一个体积不超过 X 的最大积木，依次到搭好为止，现在他想最大化积木的个数，同时在积木个数最大的情况下使 X 最大。

【输入描述】

一行一个数 m

【输出描述】

一行两个数，最多积木数以及 X 。

【样例】

tower.in	tower.out
48	9 42

【样例解释】

$X=23$ 或 42 都是 9 次， $42 = 3^3 + 2^3 + 7*1^3$

【数据范围】

30%: $m \leq 10^5$

50%: $m \leq 10^{10}$

100%: $m \leq 10^{15}$

数球

【问题描述】

小 A 有 n 个球，编号分别为 1 到 n ，小 A 每次都会从 n 个球中取出若干个球，至少取一个，至多取 n 个，每次取完再放回去，需要满足以下两个条件。

(1) 每次取出的球的个数两两不同。

(2) 每次取出的球的集合两两不包含。

包含是指，对于两次取球，对于取的数目少的那次取球的所有球都出现在取的数目多的那次取球中，例如 $\{1, 2\}$ 和 $\{1, 2, 4\}$ ， $\{1, 2\}$ 和 $\{2, 3\}$ 则不算作包含。

而小 A 现在突然想知道他最多能进行多少次这样的操作，并希望你能给出具体的取球方案。

【输入描述】

一个整数 n 。

【输出描述】

第一行一个数 k ，表示能进行的最多次数。

接下来 k 行，每行第一个整数 p ，表示这次取的球数，接下来 p 个数表示这次取的球的编号，编号只需要不同，不需要按照顺序输出，本题设有 `spj`。

对于每个测试点，每组数据第一行正确可以获得 20% 的分，如果第一行和方案均正确获得 100% 的分。

【样例】

ball.in	ball.out
4	2 1 1 2 3 4

【数据范围】

对于 30%的数据， $n \leq 7$ 。

对于 50%的数据， $n \leq 20$ 。

对于 70%的数据， $n \leq 100$ 。

对于 100%的数据， $4 \leq n \leq 1000$ 。