

Problem A. 排水系统

Time limit 1000 ms

Mem limit 524288 kB

题目描述

对于一个城市来说，排水系统是极其重要的一个部分。

有一天，小 C 拿到了某座城市排水系统的设计图。排水系统由 n 个排水结点（它们从 $1 \sim n$ 编号）和若干个单向排水管道构成。每一个排水结点有若干个管道用于汇集其他排水结点的污水（简称为该结点的汇集管道），也有若干个管道向其他的排水结点排出污水（简称为该结点的排出管道）。

排水系统的结点中有 m 个污水接收口，它们的编号分别为 $1, 2, \dots, m$ ，污水只能从这些接收口流入排水系统，并且这些结点没有汇集管道。排水系统中还有若干个最终排水口，它们将污水运送到污水处理厂，没有排出管道的结点便可视为一个最终排水口。现在各个污水接收口分别都接收了 1 吨污水，污水进入每个结点后，会均等地从当前结点的每一个排出管道流向其他排水结点，而最终排水口将把污水排出系统。

现在小 C 想知道，在该城市的排水系统中，每个最终排水口会排出多少污水。该城市的排水系统设计科学，管道不会形成回路，即不会发生污水形成环流的情况。

输入格式

从文件 `water.in` 中读入数据。

第一个两个用单个空格分隔的整数 n, m 。分别表示排水结点数与接收口数量。

接下来 n 行，第 i 行用于描述结点 i 的所有排出管道。其中每行第一个整数 d_i 表示其排出管道的数量，接下来 d_i 个用单个空格分隔的整数 a_1, a_2, \dots, a_{d_i} 依次表示管道的目标排水结点。

保证不会出现两条起始结点与目标结点均相同的管道。

输出格式

输出到文件 `water.out` 中。

输出若干行，按照编号从小到大的顺序，给出每个最终排水口排出的污水体积。其中体积使用分数形式进行输出，即每行输出两个用单个空格分隔的整数 p ， q ，表示排出的污水体积为 $\frac{p}{q}$ 。要求 p 与 q 互素， $q = 1$ 时也需要输出 q 。

样例 1

Input	Output
5 1 3 2 3 5 2 4 5 2 5 4 0 0	1 3 2 3

1 号结点是接收口，4、5 号结点没有排出管道，因此是最终排水口。
1 吨污水流入 1 号结点后，均等地流向 2、3、5 号结点，三个结点各流入 $\frac{1}{3}$ 吨污水。
2 号结点流入的 $\frac{1}{3}$ 吨污水将均等地流向 4、5 号结点，两结点各流入 $\frac{1}{6}$ 吨污水。
3 号结点流入的 $\frac{1}{3}$ 吨污水将均等地流向 4、5 号结点，两结点各流入 $\frac{1}{6}$ 吨污水。
最终，4 号结点排出 $\frac{1}{6} + \frac{1}{6} = \frac{1}{3}$ 吨污水，5 号结点排出 $\frac{1}{3} + \frac{1}{6} + \frac{1}{6} = \frac{2}{3}$ 吨污水

样例 2

见附加文件中的 [`water2.in`](file:water2.in) 与 [`water2.ans`](file:water2.ans)。

样例 3

见附加文件中的 [`water3.in`](file:water3.in) 与 [`water3.ans`](file:water3.ans)。

数据范围与提示

测试点编号	$n \leq$	$m \leq$
1 ~ 3	10	1
4 ~ 6	1000	1
7 ~ 8	10^5	1
9 ~ 10	10^5	10

对于所有测试点，保证 $1 \leq n \leq 10^5$ ， $1 \leq m \leq 10$ ， $0 \leq d_i \leq 5$ 。

数据保证，污水在从一个接收口流向一个最终排水口的过程中，不会经过超过 10 个中间排水结点（即接收口和最终排水口不算在内）。

Problem B. 字符串匹配

Time limit 1000 ms

Mem limit 524288 kB

题目描述

小 C 学习完了字符串匹配的相关内容，现在他正在做一道习题。

对于一个字符串 S ，题目要求他找到 S 的所有具有下列形式的拆分方案数：

$S = ABC$ ， $S = ABABC$ ， $S = ABAB \cdots ABC$ ，其中 A, B, C 均是非空字符串，且 A 中出现奇数次的字符数量不超过 C 中出现奇数次的字符数量。

更具体地，我们可以定义 AB 表示两个字符串 A, B 相连接，例如 $A = \text{aab}$ ， $B = \text{ab}$ ，则 $AB = \text{aabab}$ 。

并递归地定义 $A^1 = A$ ， $A^n = A^{n-1}A$ ($n \geq 2$ 且为正整数)。例如 $A = \text{abb}$ ，则 $A^3 = \text{abbabbabb}$ 。

则小 C 的习题是求 $S = (AB)^i C$ 的方案数，其中 $F(A) \leq F(C)$ ， $F(S)$ 表示字符串 S 中出现奇数次的字符的数量。两种方案不同当且仅当拆分出的 A, B, C 中有至少一个字符串不同。

小 C 并不会做这道题，只好向你求助，请你帮帮他。

输入格式

从文件 `string.in` 中读入数据。

本题有多组数据，输入文件第一行一个正整数 T 表示数据组数。

每组数据仅一行一个字符串 S ，意义见题目描述。 S 仅由英文小写字母构成。

输出格式

输出到文件 `string.out` 中。

对于每组数据输出一行一个整数表示答案。

样例 1

Input	Output
3 nnrnnr zzzaab mmlmmllo	8 9 16

对于第一组数据，所有的方案为：

1. $A = n, B = nr, C = nnr$ 。
2. $A = n, B = nrn, C = nr$ 。
3. $A = n, B = nrnn, C = r$ 。
4. $A = nn, B = r, C = nnr$ 。
5. $A = nn, B = rn, C = nr$ 。
6. $A = nn, B = rnn, C = r$ 。
7. $A = nnr, B = n, C = nr$ 。
8. $A = nnr, B = nn, C = r$ 。

样例 2

Input	Output
5 kkkkkkkkkkkkkkkkkkkk llllllllllllrrllrr ccccccccccccxcxxcc cccccccccccccaababa gggggggggggggggbaabab	156 138 138 147 194

样例 3

见附加文件中的 [string3.in](file:string3.in) 与 [string3.ans](file:string3.ans)。

样例 4

见附加文件中的 [string4.in](file:string4.in) 与 [string4.ans](file:string4.ans)。

数据范围与提示

测试点编号	$ S \leq$	特殊限制
1 ~ 4	10	无
5 ~ 8	100	无

测试点编号	$ S \leq$	特殊限制
9 ~ 12	1000	无
13 ~ 14	2^{15}	S 中只包含一种字符
15 ~ 17	2^{16}	S 中只包含两种字符
18 ~ 21	2^{17}	无
22 ~ 25	2^{20}	无

对于所有测试点，保证 $1 \leq T \leq 5, 1 \leq |S| \leq 2^{20}$ 。

Problem C. 移球游戏

Time limit 1000 ms

Mem limit 524288 kB

题目描述

小 C 正在玩一个移球游戏，他面前有 $n + 1$ 根柱子，柱子从 $1 \sim n + 1$ 编号，其中 1 号柱子、2 号柱子、 \dots 、 n 号柱子上各有 m 个球，它们自底向上放置在柱子上， $n + 1$ 号柱子上初始时没有球。这 $n \times m$ 个球共有 n 种颜色，每种颜色的球各 m 个。

初始时一根柱子上的球可能是五颜六色的，而小 C 的任务是将所有同种颜色的球移到同一根柱子上，这是唯一的目标，而每种颜色的球最后放置在哪个柱子则没有限制。

小 C 可以通过若干次操作完成这个目标，一次操作能将一个球从一根柱子移到另一根柱子上。更具体地，将 x 号柱子上的球移动到 y 号柱子上的要求为：

1. x 号柱子上至少有一个球；
2. y 号柱子上至多有 $m - 1$ 个球；
3. 只能将 x 号柱子最上方的球移到 y 号柱子的最上方。

小 C 的目标并不难完成，因此他决定给自己加加难度：在完成目标的基础上，使用的操作次数不能超过 820000。换句话说，小 C 需要使用至多 820000 次操作完成目标。

小 C 被难住了，但他相信难不倒你，请你给出一个操作方案完成小 C 的目标。合法的方案可能有多种，你只需要给出任意一种，题目保证一定存在一个合法方案。

输入格式

从文件 `ball.in` 中读入数据。

第一行两个用空格分隔的整数 n, m 。分别表示球的颜色数、每种颜色球的个数。

接下来 n 行每行 m 个用单个空格分隔的整数，第 i 行的整数按自底向上的顺序依次给出了 i 号柱子上的球的颜色。

输出格式

输出到文件 `ball.out` 中。

本题采用自定义校验器 (Special Judge) 评测。

你的输出的第一行应该仅包含单个整数 k ，表示你的方案的操作次数。你应保证 $0 \leq k \leq 820000$ 。

接下来 k 行每行你应输出两个用单个空格分隔的正整数 x, y ，表示这次操作将 x 号柱子最上方的球移动到 y 号柱子最上方。你应保证 $1 \leq x, y \leq n + 1$ 且 $x \neq y$ 。

样例 1

Input	Output
2 3 1 1 2 2 1 2	6 1 3 2 3 2 3 3 1 3 2 3 2

柱子中的内容为：按自底向上的顺序依次给出柱子上每个球的颜色。

操作	1 号柱子	2 号柱子	3 号柱子
初始	1 1 2	2 1 2	
1 3	1 1	2 1 2	2
2 3	1 1	2 1	2 2
2 3	1 1	2	2 2 1
3 1	1 1 1	2	2 2
3 2	1 1 1	2 2	2
3 2	1 1 1	2 2 2	

样例 2

见附加文件中的 [\[ball2.in\]](#)(file:ball2.in) 与 [\[ball2.ans\]](#)(file:ball2.ans)。

样例 3

见附加文件中的 [\[ball3.in\]](#)(file:ball3.in) 与 [\[ball3.ans\]](#)(file:ball3.ans)。

数据范围与提示

测试点编号	$n \leq$	$m \leq$
1 ~ 2	2	20
3 ~ 5	10	20
6 ~ 8	50	85
9 ~ 14	50	300
15 ~ 20	50	400

对于所有测试点，保证 $2 \leq n \leq 50$ ， $2 \leq m \leq 400$ 。

校验器

为了方便选手测试，在下发文件中我们下发了 `checker.cpp` 文件，选手可以编译该程序，并使用它校验自己的输出文件。但请注意它与最终评测时所使用的校验器并不完全一致。你也不需要关心其代码的具体内容。

编译命令为：`g++ checker.cpp -o checker -std=c++11`。

`checker` 的使用方式为：`checker <inputfile> <outputfile>`，参数依次表示输入文件与你的输出文件。

若你输出的数字大小范围不合法，则校验器会给出相应提示。若你的输出数字大小范围正确，但方案错误，则校验器会给出简要的错误信息：

1. `A x`，表示进行到第 x 个操作时不合法。
2. `B x`，表示操作执行完毕后第 x 个柱子上的球不合法。

若你的方案正确，校验器会给出 `OK`。

Problem D. 微信步数

Time limit 1000 ms

Mem limit 524288 kB

题目描述

小 C 喜欢跑步，并且非常喜欢在微信步数排行榜上刷榜，为此他制定了一个刷微信步数的计划。

他来到了一处空旷的场地，处于该场地中的人可以用 k 维整数坐标 (a_1, a_2, \dots, a_k) 来表示其位置。场地有大小限制，第 i 维的大小为 w_i ，因此处于场地中的人其坐标应满足 $1 \leq a_i \leq w_i$ ($1 \leq i \leq k$)。

小 C 打算在接下来的 $P = w_1 \times w_2 \times \dots \times w_k$ 天中，每天从场地中一个新的位置出发，开始他的刷步数计划（话句话说，他将会从场地中每个位置都出发一次进行计划）。

他的计划非常简单，每天按照事先规定好的路线行进，每天的路线由 n 步移动构成，每一步可以用 c_i 与 d_i 表示：若他当前位于 $(a_1, a_2, \dots, a_{c_i}, \dots, a_k)$ ，则这一步他将会走到 $(a_1, a_2, \dots, a_{c_i} + d_i, \dots, a_k)$ ，其中 $1 \leq c_i \leq k$ ， $d_i \in \{-1, 1\}$ 。小 C 将会不断重复这个路线，直到他走出了场地的范围才结束一天的计划。（即走完第 n 步后，若小 C 还在场内，他将回到第 1 步从头再走一遍）。

小 C 对自己的速度非常有自信，所以他并不在意具体耗费的时间，他只想知道 P 天之后，他一共刷出了多少步微信步数。请你帮他算一算。

输入格式

从文件 `walk.in` 中读入数据。

第一行两个用单个空格分隔的整数 n, k 。分别表示路线步数与场地维数。

接下来一行 k 个用单个空格分隔的整数 w_i ，表示场地大小。

接下来 n 行每行两个用单个空格分隔的整数 c_i, d_i ，依次表示每一步的方向，具体意义见题目描述。

输出格式

输出到文件 `walk.out` 中。

仅一行一个整数表示答案。答案可能很大，你只需要输出其对 $10^9 + 7$ 取模后的值。

若小 C 的计划会使得他在某一天在场地中永远走不出来，则输出一行一个整数 -1 。

样例 1

Input	Output
3 2 3 3 1 1 2 -1 1 1	21

从 (1, 1) 出发将走 2 步，从 (1, 2) 出发将走 4 步，从 (1, 3) 出发将走 4 步。从 (2, 1) 出发将走 2 步，从 (2, 2) 出发将走 3 步，从 (2, 3) 出发将走 3 步。从 (3, 1) 出发将走 1 步，从 (3, 2) 出发将走 1 步，从 (3, 3) 出发将走 1 步。共计 21 步。

样例 2

Input	Output
5 4 6 8 6 5 3 1 2 1 1 1 2 1 2 -1	10265

样例 3

见附加文件中的 [[walk3.in](#)](file:walk3.in) 与 [[walk3.ans](#)](file:walk3.ans)

样例 4

见附加文件中的 [[walk4.in](#)](file:walk4.in) 与 [[walk4.ans](#)](file:walk4.ans)

数据范围与提示

测试点编号	$n \leq$	$k \leq$	$w_i \leq$
1 ~ 3	5	5	3
4 ~ 6	100	3	10
7 ~ 8	10^5	1	10^5
9 ~ 12	10^5	2	10^6
13 ~ 16	5×10^5	10	10^6

测试点编号	$n \leq$	$k \leq$	$w_i \leq$
17 ~ 20	5×10^5	3	10^9

对于所有测试点，保证 $1 \leq n \leq 5 \times 10^5$ ， $1 \leq k \leq 10$ ， $1 \leq w_i \leq 10^9$ ， $d_i \in \{-1, 1\}$ 。