

问题 A. 魔法计算机

输入文件：标准输入 标准输出 1

输出文件：秒 256 兆字节

时限：

内存限制：

Bob 有一台神奇的电脑,它有无限的 USB 端口。但是这台电脑有个问题 只能读写最近插入的两个U盘。

最初,Bob 有 n 个 U 盘,每个 U 盘都有一个唯一的文件。他要将这 n 个U盘全部插入到自己的电脑中。由于他的电脑是旧的,他必须通过以下方式插入U盘：

1. 选择2个未插入的U盘,插入电脑。
2. 上一步插入的两个U盘会互相共享文件,也就是说两个U盘之后都会有第一个U盘或者第二个U盘的所有文件。
- 3.如果 n 个U盘都在电脑中,则任务完成。否则,选择一个 USB 磁盘
电脑,拔下它并转到步骤 1。

Bob 想知道 n 的最小数量,使得有一个操作序列最终在每个 USB 磁盘至少有 k 个不同的文件。

输入

只有一个数 k ($2 \leq k \leq 100000$) ,也就是说最后每个U盘至少拥有 k 个不同的文件。

输出

需要输出一个数字 n ,表示模块998244353中最少的U盘数。

例子

标准输入	标准输出
3个	4个
5个	16
1000	510735315

问题 B. Chevonne 的游戏

输入文件：标准输入 标准输出 5 秒

输出文件：512 兆字节

时限：

内存限制：

你想和 Chevonne 一起玩游戏吗？

Chevonne是一个善良的女巫。她有许多神奇的珍珠,颜色或白或黑。珍珠排成一行,编号从 1 到 n。 Chevonne 可能会在您一些疑问的同时改变一些珍珠的颜色。凡能正确回答所有问题的人,都会得到她的祝福,获得“Accepted”。

具体来说,Chevonne 会通过以下两种形式执行 q 动作：

- MLR Chevonne 使用魔法改变颜色（白变黑,黑变白）
珍珠编号从L到R。
- QLR Chevonne 给你一个查询,最少需要多少次才能取走所有编号为L到R的珍珠。你只能通过以下方式取走珍珠 :选择几个连续的珍珠（概念是类似于子串）,并且每两个相邻的珍珠颜色不同（选择一个珍珠总是可以的）。把这些珍珠拿走,把剩下的珍珠（如果有的话)合并成一个整体。

输入第一

行包含两个整数 n (1 ≤ n ≤ 106) 和 q (1 ≤ q ≤ 106)。

第二行是一个长度为 n 的二进制字符串 s,第 i 个字符代表编号为 i 的珍珠的颜色（0 代表白色,1 代表黑色）。

接下来是 q 行。每行包含一个字符和两个整数L,R （1≤L≤R≤n）,代表Chevonne将执行的动作。

输出

对于每个查询,请打印一行单个数字代表相应的答案
询问。

例子

标准输入	标准输出
8 5	2个
10001101	3个
问 1 3	1个
问 1 8	2个
米 3 5	
问 1 3	
问 1 8	

笔记

对于第一个查询,我们需要计算出最少需要多少次才能拿走前三颗珍珠 “100”。

答案是2。一个可行的方案是：

1. 取走前两颗珍珠 “10” ； 2. 拿走剩下的一颗珍珠 “0”。

问题 C. 指南针

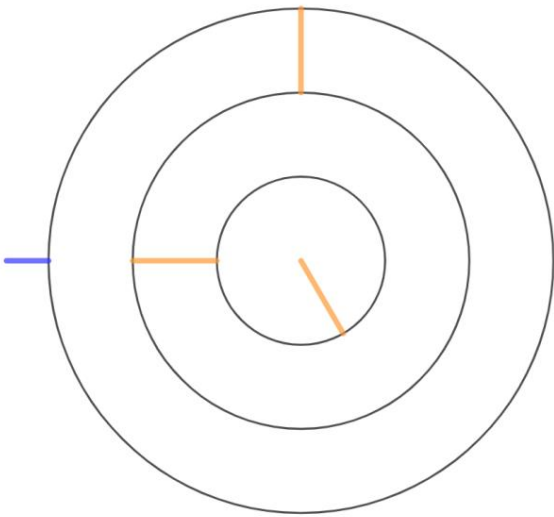
输入文件：标准输入 标准输出 1

输出文件：秒 256 兆字节

时限：

内存限制：

崩坏:星轨是一款玩家到达仙洲罗府时可能会遇到名为航海罗盘的谜题的游戏。这个拼图由三层组成,每层都有一条橙色线。



这个谜题旨在通过顺时针旋转图层将所有橙色线与蓝色线对齐。随着层的旋转,橙色线会移动位置,但蓝色线保持固定。由于天文原因,每一层在一步中可以旋转的程度是预先确定的。例如,如果第 1 层每步可以旋转 60°,则需要 6 步才能完成完整的旋转。

Wor 先生卡在了这个谜题的最后一层,玩家只能在每一步中同时旋转任意两层。给定橙色线的初始位置和每层完成完整旋转所需的步数,请找出对齐所有线所需的最少步数。

也就是说,给你6个整数 $x_0、x_1、x_2、y_0、y_1、y_2$,请找出三个非负整数 $t_0、t_1、t_2$,使得 $x_0 + t_1 + t_2 \equiv 0(mod y_0)$, $x_1 + t_0 + t_2 \equiv 0(mod y_1)$, $x_2 + t_0 + t_1 \equiv 0(mod y_2)$,并最小化 $t_0 + t_1 + t_2$ 。

输入

第一行包含一个整数 $T(1 \leq T \leq 10)$,表示测试用例的数量。
然后在接下来的 T 行中,每行包含 6 个整数 $x_0, x_1, x_2, y_0, y_1, y_2(\forall i, 0 \leq x_i < y_i \leq 2000)$ 。

输出

对于每个测试用例,打印最小值 $t_0 + t_1 + t_2$ 。如果没有解,打印-1。

例子

标准输入	标准输出
3个 1 1 1 2 3 4 1 1 0 2 3 4 1 1 1 4 7 5	3个 5个 9

问题 D. 大流行

输入文件：标准输入 标准输出 2

输出文件：秒 512 兆字节

时限：

内存限制：

Kanade 所在的大学最近因大流行病被迫关闭。学生们被锁在宿舍楼里,不准外出。但幸运的是,学校为他们准备了丰盛的免费餐食。同学们在享受周到服务的同时,也热心志愿服务,为学校的工作贡献自己的微薄之力。奏也不例外。

Kanade 的工作是给 n 个房间的学生分发盒饭,房间编号从 1 到 n ,这些房间从左到右连续排列在一条走廊上,每个房间有 4 个学生。学校提供 m 种不同的盒饭,每一种都供应充足（意思是无限量）。

奏决定将盒饭从左到右分发到各个房间。具体来说,第 i 次分发是从盒饭箱中随机取出 4 份盒饭,送到第 i 个房间。然而几天后,奏觉得在大楼里来回奔波、挨个房间敲门实在是太累了,于是他想出了一个更轻松的方法。

顺序依然是从左到右,但是第 i 次分配是从箱子里随机取出 $4k_i$ 盒饭,从左到右第一个没有 的房间开始敲连续 k_i 个房间的门 没有分发。然后,他会将 $4k_i$ 盒饭放在地板上,供 k_i 室的学生选择。这个过程会一直持续到 n 个房间都分配完为止。每个 k_i 都是一个正整数,奏可以决定,但是他不想 k_i 过大造成不必要的混乱,所以规定 $k_i \leq K$ 。

Kanade 想知道有多少种不同的计划以这种方式分发盒饭。当且仅当分发次数不同,第 i 次分发的 k_i 不同,或者第 i 次分发中取出的任意一种盒饭的数量不同时,才认为两个计划不同。

由于这个数字可能太大,输出它模 998244353。

输入

有 只有一行,包含 n, m, K ($1 \leq K \leq n \leq 105$ 个盒饭),以及 Kanade 三 积极的 integer numbers $1 \leq m$ 在每次分配中可以分配的房间上限。 , ≤ 105) 房间的数量,类型的数量

输出

输出单个数字 不同分配计划的数量对 998244353 取模。

例子

标准输入	标准输出
3 1 2	3个
3 2 1	125
10 4 3	559774277

笔记

请注意,Kanade 想知道的是不同分配计划的数量,这意味着他不关心学生如何分配第 i 次分配中的 $4k_i$ 盒饭。

问题 E. 以太网

输入文件：标准输入 标准输出 1
输出文件：秒 256 兆字节
时限：
内存限制：

Wor先生家里的路由器又坏了。经过一些维修后,他需要将以太网电缆重新连接到路由器。

已知Wor先生的路由器有n个端口,有n根网线。然后,Mr. Wor 将依次将第 i 条以太网电缆插入第 i 个端口,对于所有 i 从 1 到 n。

但是,由于Wor先生的神秘举动,前m根电缆并没有插入到它们指定的端口中,而是随机均匀地插入到m个端口中。其余线缆,在插入线缆i时,如果端口i没有被占用,则将线缆i插入端口i。否则,电缆 i 将被插入到随机选择的未占用端口。

Wor 先生想知道最后一根电缆,即电缆 n,插入端口 n 的概率。

输入

只有一行包含两个整数n,m (1≤n≤10,0≤m≤n) ,分别表示电缆的数量和随机插入的电缆的数量。

输出

在一行中打印概率。如果相对或绝对误差小于或等于10−6 ,则认为您的答案是正确的。

例子

标准输入	标准输出
3 1	0.5000000000

问题 F. 文件夹

输入文件：标准输入 标准输出 1

输出文件：秒 256 兆字节

时限：

内存限制：

一台电脑中有n个文件夹,索引从1到n。索引为 1 的文件夹是根文件夹,每个文件夹可能包含多个子文件夹,形成树状结构。您可以执行多个“剪切”操作,每个操作将一个源文件夹剪切到一个目标文件夹中。注意目标文件夹不能和源文件夹相同,也不能是源文件夹的子文件夹,子文件夹的子文件夹等等。显然,根文件夹是不能切的,永远是根文件夹。找出至少应该执行多少次“剪切”操作才能使每个文件夹最多包含一个子文件夹。

输入第一

行包含一个整数 n ($1 \leq n \leq 105$) 文件夹数。each的第二行, p_i ($1 \leq p_i \leq n$) p_i 表示测试用例的索引包含n-1个空格分隔的整数 p_2, p_3, \dots 文件夹i的父文件夹。

输出

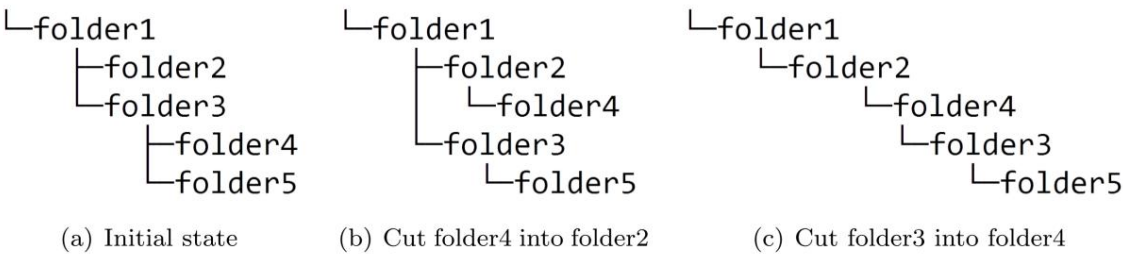
输出执行的“剪切”操作的最少次数。请注意,可能不需要执行“剪切”操作。在这种情况下请输出 0。

例子

标准输入	标准输出
5个 1 1 3 3	2个

笔记

这是第一个样本的解释。



问题 G. 重力

输入文件：标准输入 标准输出 1
输出文件：秒 256 兆字节
时限：
内存限制：

约瑟夫今天的任务是收集小行星来建造新的空间站。

约瑟夫所在的宇宙是一个二维平面。小行星位于由 P_n 表示的 n 个点。它们的质量可以认为是相等的。第一天,他将收集一些小行星 P_1 、 P_2 、 A 和 B 表示。... , 并把它们带到它们的重心位置。第二天,他将对剩余的小行星进行同样的处理。然后,他将在两个重心处建造两个新站,用点

注意到建造一个空间站至少需要一颗小行星,所以约瑟夫每天至少要收集一颗小行星。

$O(0, 0)$ 点上有一个原始空间站。三个空间站的覆盖区域由三角形 ABO 定义。约瑟夫想要尽可能地最大化这个区域。

请帮他找到一个最优的收集方案,以获得最大的覆盖面积。

Input第

一行包含一个整数 n ($2 \leq n \leq 105$), 表示小行星的数量。... , y_i ($-104 \leq x_i, y_i \leq 104$), 表示有 在接下来的 n 行中,第

i 行包含两个整数 x_i 一颗小行星在点 $P_i = (x_i, y_i)$

输出

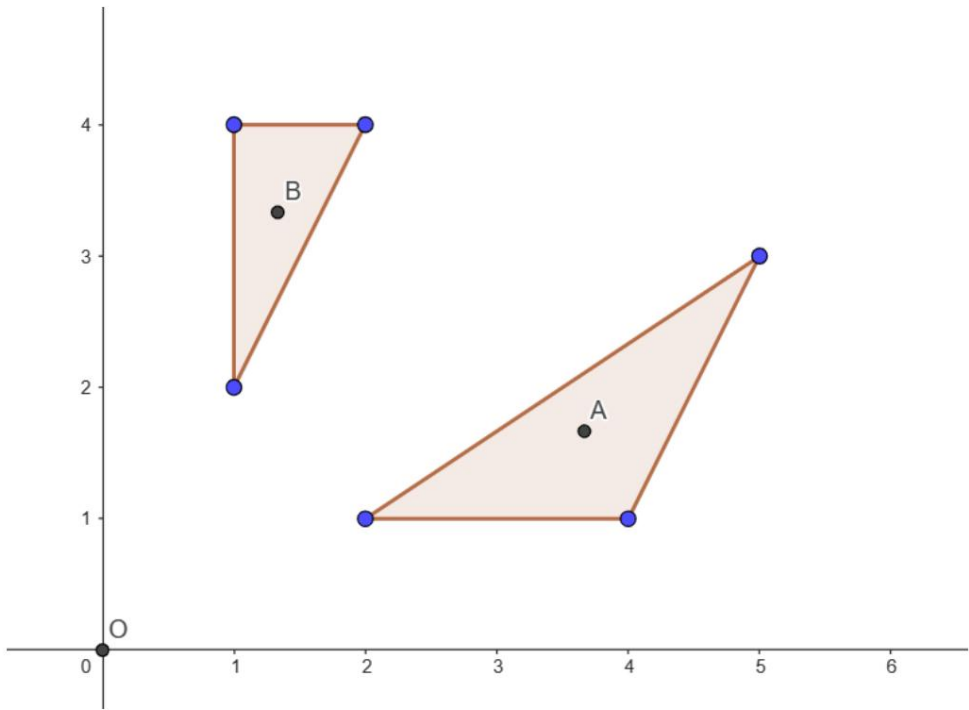
在一行中打印最大覆盖区域。如果相对或绝对误差小于或等于 10^{-9} , 则认为您的答案是正确的。

例子

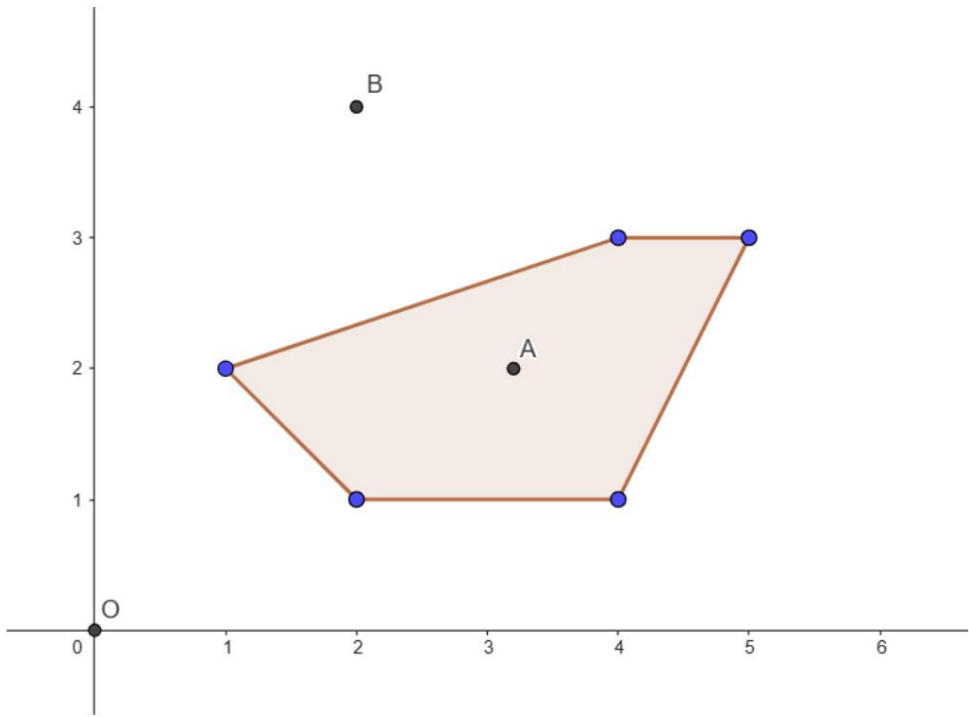
标准输入	标准输出
<pre>6 2 1 1 2 4 1 1 4 5 3 2 4</pre>	5.0000000000
<pre>6 2 1 1 2 4 1 4 3 5 3 2 4</pre>	4.4000000000

笔记

下图描述了第一个示例。一个最优的采集方案是第一天采集第1、3、5号小行星,第二天采集第2、4、6号小行星。重心分别为A和B。三角形ABO的面积为5.0,最大化。



下图描述了第二个示例。



问题 H. KingZ

输入文件：标准输入 标准输出 1
输出文件：秒 256 兆字节
时限：
内存限制：

Wor先生是《星际争霸》等RTS游戏的资深玩家。不过就在近日,暴雪宣布他们将退出中国市场。作为暴雪娱乐的狂热爱好者,Wor 深感沮丧,不得不寻找替代品。在这种情况下,《KingZ》这款游戏吸引了Wor的目光。

KingZ 是一款桌面即时战略游戏,这意味着两名玩家同时对战并做出决定。现在为您提供有关此游戏的一些基本知识。



基本游戏规则

如图所示,只有两名玩家 (红色和蓝色)在 10 × 10 大小的棋盘上玩这个游戏。这是一个回合制游戏,这意味着玩家在每一轮中做出决定。每个玩家都占领了一些有执勤守备的场地,也就是有一些占领场地上的部队。驻军是指战场上的部队人数。在每一轮比赛中,一些援军将增援被占领的场地。根据不同类型的领域,将增加不同数量的增援部队。在这个游戏中,双方玩家在每一轮游戏中同时做出决定和制定策略。最后,谁先击败对面的核心,谁就是胜利者。

地图和物品地图由

10×10 块组成,我们称之为“场地”。每个场地有四种类型:Core、Keep、Lawn 和 Wall。每个字段都有自己的坐标,左上角为 (0, 0),右下角为 (10, 10)。注意每个玩家只有一个核心。

驻军和职业

辞典:附属的意思是属于。

除了城墙外,每个领域都可以被驻军占领。最初,草坪没有驻军,也没有任何玩家占据,Keep 有中立驻军,每个玩家只占据有 10 个驻军的核心。请注意,这只是初始情况,会随着游戏的进行而改变。

一旦某个场地被某人的驻军占领,该场地的控制权就属于该玩家。

一个令人费解的情况是,有时一个领域在一轮中没有驻军,这个领域的隶属关系

将保留,这意味着隶属关系与上一轮占据该领域的玩家保持不变。

战斗

决定完成后,每一轮结束时,如果某个阵地有不同隶属的驻军,则进行战斗。

战斗将按照以下步骤进行:

首先,如果蓝色和红色都有驻军附属,他们将首先通过减去驻军数量进行战斗。然后还有驻军的强者继续战斗。

其次,剩余的驻军与中立部队的战斗也是通过减去他们的驻军数量来进行的。

最后,绝对最多只有一个驻军的隶属存活下来并占领该领域。

形式上来说,我们定义该区域有 x 个隶属于蓝色的驻军, y 个隶属于红色的驻军, z 个驻军是中立的,现在执行以下步骤:

米=分钟 (x,y)

$x=xm$

$y=ym$ if

($x>0$) then begin

米=分钟 (x,z)

$x=xm$

$z=z$ 变量

结尾

else if ($y>0$) 然后开始

$m=\min(y,z)$

$y=y$ 输入

$z=z$ 变量

结尾

加强

每回合结束后,核心驻军数量会增加2个。

每回合结束后,如果要塞被任何玩家占领,则要塞内的驻军数量会增加1。请注意,仍处于中立状态的要塞将不会得到增援。

每8k回合结束后,任意玩家占领的草坪上的驻军数量增加1。

决策 (操作)

每个决策,或者我们称之为操作,可以翻译成五元组 (ux, uy, x, vx, vy),也就是说驻守在 (ux, uy)的 x 个部队会直接移动到 (vx, vy)成本。你必须保证 $x \leq \text{rounds number} + \text{dis}(u, v)$, $\text{dis}(u, v) = |ux - vx| + |uy - vy|$, $0 \leq x \leq \text{garrison}_{ux,uy}$ 。

胜利或失败首先击败对方核

心的玩家将获胜。

额外限制

请注意,此问题中的这些额外限制很重要。

在这个问题中,可以在一个回合中进行无限次操作。并且你必须确保在所有操作完成后,占领领域的每个驻军数量都大于零。

虽然可以进行无限次操作,但四元组 (ux, uy, vx, vy)在操作列表中只会存在一次。

主要问题

现在告诉你现在的情况,在你开始操作之前除了加固什么都不会做,也就是说在红蓝玩家开始操作之前,两个玩家都什么都不做,我们称之为等待回合.你的任务是计算最少的等待轮数,通过等待这些轮数,你可以通过无限次操作在一个回合内占领除墙外的所有场地.为了简化问题,我们可以很容易地认为 opposite 自始至终什么都不做。

请注意,如果回合数超过 300,那么我们判断游戏为平局并输出 -1 。

输入

包含两个 10×10 整数矩阵 $a_{i,j}$, $c_{i,j}$, 第一个输入是整数矩阵 $a_{i,j}$ ($-1 \leq a_{i,j} \leq 256$),第二个输入是整数矩阵 $c_{i,j}$ ($c_{i,j} \in \{0,1,2,3,4,5,6,7\}$)。

矩阵 $a_{i,j}$ 表示:

$a_{i,j}$	意义
-1	城墙
0	没有驻军 >0 被占领并
且有 $a_{i,j}$ 驻军部队	

矩阵 $c_{i,j}$ 的意思是: $c_{i,j}$ 的意思是0

没有从属关系	
1*	我们的核心
	我们保留
	我们的草坪
2 3 4	敌人核心 5 敌人
据点	6 敌人草坪 7 附属
据点	

输出

只有一个整数ans表示最少等待轮数,通过等待这些轮数,您可以通过进行无穷大操作在单轮中占据除墙以外的所有领域。

注意到如果回合超过 300,那么我们判断游戏为平局并输出 -1 。

例子

标准输入7 5 -1 3 15	标准输出
12 8 -1 47 0 4 2 -1 61 2 39 59 2 3 -1 2 11 3 1 -1 7 1 3 3 56 3 -1 1 74 -1 -1 2 3 3 8 0 -1 2 -1 7 6 2 6 60 7 2 60 2 4 5 6 -1 6 -1 6 2 9 2 4 -1 -1 7 4 6 -1 4 5 6 10 2 4 5 -1 3 16 0 2 -1 2 2 4 3 3 3 3 -1 0 2 0 4 7 -1 10 10 5 4 10 -1 0 3 3 6 0 3 2 3 3 0 7 0 3 3 0 6 3 1 3 3 3 0 3 3 5 6 0 3 3 3 3 7 3 0 6 7 0 0 3 3 3 3 0 0 6 0 3 3 3 3 7 3 6 7 6 6 3 3 0 3 0 3 6 6 6 6 0 0 7 3 0 3 7 6 6 6 6 0 6 5 0 3 0 6 6 6 4 6 6 0 0 3 0 7 0 6 6 7 6 0 0 3	-1
0032400000 0044400000 3249100000 0352100000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0033300000 0033300000 3331300000 0333300000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000	16

问题一、俱乐部

输入文件：标准输入标准输出1秒
输出文件：
时限：
内存限制：512 兆字节

4月,春意盎然,一年一度的盛事 “百会大战”将在吉大举行。

JLU 中有 n 个俱乐部。“百会大战”当天,各会社将举办自己的社团游戏,参赛者可参与游戏赢取徽章,集齐各类徽章的玩家还可获得神秘大奖!

作为JLU俱乐部协会的成员和赛事的组织者,Paulliant设计了 m 枚徽章,并为每个俱乐部分配了其中一枚。考虑到参与者不知道每个俱乐部的徽章类型,他们将随机参加一次一个俱乐部 (可以是之前选择的俱乐部)的游戏并获得其徽章,直到集齐所有类型的徽章。Paulliant 想知道如何将 m 个徽章分配给 n 个俱乐部,以便最大限度地减少参与者玩游戏的预期次数。输出这个期望值。

请注意,相同的徽章可以分配给不同的俱乐部,但每个徽章必须至少分配给一个俱乐部。

Input只

有一行包含两个正整数 n, m ($1 \leq n \leq 105$ $1 \leq m \leq 5000, m \leq n$) JLU 俱乐部的数量,以及 Paulliant 设计的徽章数量,由一个空格分隔。

输出

输出一个数字 参与者玩游戏的最小预期数量。如果绝对或相对误差不超过 10^{-6} ,您的答案将被视为正确。

例子

标准输入	标准输出
3 2	3.5000000000
111 7	18.1658637604

问题 J. XOR 字符串

输入文件：标准输入 标准输出 1
输出文件：秒 256 兆字节
时限：
内存限制：

给你一个字符串 S , 由 n 个小写英文字母 $S_0, S_1, S_2, \dots, S_{n-1}$ 组成。

字符串 S 中的每个字符都有一个值 V_i 。

定义 $U = \{x | x \in \mathbb{Z}, 0 \leq x \leq n-1\}$ 。

如果存在满足以下要求的非空集合 P , 则称字符串 t ($|t| \leq n$) 是合格的：

- $P \subseteq U$
- $\forall p \in P, \forall i \in [0, |t| - 1], t_i = S_{(p+i) \bmod n}$
- $\forall p \in U \setminus P, \exists i \in [0, |t| - 1], t_i = S_{(p+i) \bmod n}$
- $\bigoplus_{p \in P} V_p = 0$, 这意味着对于 P 中的所有元素 p , V_p 的异或和等于 0。

请计算最长的合格字符串 t 的长度。

Input 第

一行包含一个长度为 n ($1 \leq n \leq 105$) 的字符串 S , 由小写英文字母组成。 V_{n-1} ($0 \leq V_i < 2 \cdot 10^9$)。第二行包含 n 个整数, V_0, V_1, V_2, \dots ,

输出

输出一个整数, 即最长合格字符串 t 的长度。如果没有符合条件的字符串, 则输出 0。

例子

标准输入	标准输出
父亲 1 1 2 3 2 4	3个

问题 K. 回合制游戏

输入文件：标准输入标准输出1秒

输出文件：

时限：

内存限制：256 兆字节

在回合制游戏中,玩家控制角色打败怪物。现在,您需要根据规则找到最优策略。

在这个游戏中,玩家拿着一个血点为A的盾牌。每个怪物有B血点,当血量小于或等于0时,一个怪物死亡。在每一轮中,玩家选择一个怪物（任何一个怪物)并攻击它,然后每个活着的怪物都会攻击玩家一次。为了简化问题,游戏中的所有攻击只会让目标掉1点血。

现在玩家将进行 n 场战斗。在第 i 场战斗中,有ai 个怪物。请注意,当一场战斗中的所有怪物死亡时,新的战斗将开始新一轮（这意味着玩家在新的战斗中首先攻击）。

一个叫Seele的角色可以帮助玩家更高效的击杀怪物。如果玩家在 Seele 的帮助下在一次攻击中杀死了一只怪物,他/她可以立即进行另一次攻击。对于每场战斗的第 i 次攻击（对于每场战斗,此计数从 1 开始）， bi = 1 表示 Seele 会提供帮助， bi = 0 表示她不会提供帮助。保证当 i 大于 m 时， bi = 0。

如果剩余护盾的血量小于或等于 0,则玩家输。为了避免这种情况,请找到一个策略来最大化剩余盾牌的血量。

输入

第一行包含两个整数A（1≤A≤105)和B（1≤B≤20）,盾牌和怪物的血点。

第二行包含两个整数n（1≤n≤105)和m（1≤m≤2000）,战斗次数和Seele的帮助上限。

第三行有n个整数,第i个整数是ai（1≤ai≤100）。

第四行有m个整数,第i个整数为bi（0≤bi≤1,0≤Σ bi≤20）。

输出

如果玩家输了,输出“LOSE”。否则,输出一个整数,即剩余盾牌的最大血量。

例子

标准输入	标准输出
100 2 3 6 2 3 4 0 0 0 1 1 1	75
10 3 1 3 1 1 0	8个

问题 L. Subxor

输入文件：标准输入 标准输出 1

输出文件：秒 256 兆字节

时限：

内存限制：

给定一个数组 a_1, a_2, \dots, a_n , 你需要回答 Wor 先生提出的多个问题, 他有兴趣了解数组的特定方面。

每个查询由两个整数 l 和 r ($1 \leq l \leq r \leq n$) 组成。您需要找到 a_l, a_{l+1}, \dots, a_r , 使得其元素的异或和小于或等于给定值 K 。

具体来说, 需要找到一对 u, v 并最大化 $v - u + 1$ 的值, 条件如下:

$$l \leq u \leq v \leq r$$
$$a_u \oplus a_{u+1} \oplus \dots \oplus a_v \leq K$$

请注意, K 的值是预先确定的, 并且对所有查询保持不变。如果不存在有效对 u, v , 则输出 0。

请为 Wor 先生编写一个程序, 以正确回答所有问题。

输入

第一行包含 2 个整数 n, K ($1 \leq n \leq 2 \times 10^4, 0 \leq K < 2^{31}$)。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i < 2^{31}$), 表示数组。

($1 \leq q \leq 2 \times 10^4$) , 表示查询次数。

在接下来的 q 行中, 每行包含 2 个整数 l, r ($1 \leq l \leq r \leq n$), 描述一个查询。

输出

对于每个查询, 在一行中打印一个整数, 表示子数组的最大长度。如果这样的子数组不存在, 则打印 0。

例子

标准输入	标准输出
5 1 1 2 3 4 5 4个 1 5 2 4 3 4 3 5	5个 2个 0 2个