

2021 蓝桥杯热身赛 #2

试题册

华南师范大学软件协会香农先修班
华南师范大学软件学院 ICPC/CCPC 集训队

题目列表

异世界的历法(5分)

异世界的文章(5分)

异世界的融资(10分)

异世界的地图(10分)

异世界的招聘(15分)

异世界的剧情分支(15分)

异世界的人偶训练(20分)

异世界的稳定测试(20分)

异世界的模块合并(25分)

异世界的魔法优化(25分)

重要提示 Java 和 Python 程序在题目所标时间空间限制基础上有 2 秒的额外运行时间和 64 MB 的额外空间

A. 异世界的历法(5分)

1 秒, 128 MB

果冻通过千层塔试炼，成功来到了名为蓝星的剑与魔法的中世纪异世界。他发现，异世界的历法与原世界的完全不一样。在异世界里，1 年有 10 月，1 月有 10 天，1 天有 40 个时间单位(异世界小时)。而一个异世界小时等于原世界的 55 分钟。

果冻在异世界时间的 580 年 3 月 7 日 15 时整穿越到了这里。他预计自己需要花费原世界 1080 天的时间才能完全适应这个世界。求经过该时间后在异世界时间的哪年哪月哪日。设时间为 y 年 m 月 d 日，你应当以一个空格为间隔依次输出 y, m, d 。

例如，经过原世界 20 天后，异世界时间为 580 年 4 月 10 日，应当输出 `580 4 10`。

输入

无

输出

你的答案

B. 异世界的文章(5分)

1 秒, 128 MB

这学习，多是一件美事。果冻正在学习异世界的语言文字。他读到一篇有趣的文章。定义字符串子序列是不一定连续的子字符串，如对字符串 `yisaigai`，`sai`，`yii` 是子序列，`gs`，`ha` 不是子序列。果冻尤其喜欢以 `a` 开头，`k` 结尾的三字词语，如 `afk`，`auk`。果冻想找到文章的所有子序列里有多少个子序列是这样的三字词语。

例如，假设文章为 `idlikelangiaoak`，那么其所有子序列中有 8 个 `a*k` 三字词。

对于如下文章，求该文章的所有子序列里有多少个是这样的三字词语。(为避免你处理标点符号，我们已去除掉所有排版和标点符号)

icpcfinalisttizhuxianshengninhaoninhaoninhaoyoushenmeshiwomennengbangdaonitizhupazhuochuanqihuwoyaoshuode
shinimenqianwanbiehaipaicpcfinalistwomenshijinpaixuanshouwomenbuhuipaninqingshuotizhuwogangcaizaileetcode
zhousaibaolingicpcfinalistnoiausuranqijingzhanshuhouyangicpcfinalistleetcodeshinayiweitizhubushinayiwei
jishinagehenduo henduosuanfanantideleetcodenoiauhuachunoiptizhubushizhongxueshengdadeshidaxueshengdadenoia
u huachuicpctizhubutazaiwangluoshangdadbushixianchangsaioiauhuachudeforcestizhutinemeiyingwendetashizho
ngwendenoiauhuachutopcodertizhuzheicpcfinalistduoguo zhihuachunanchangyaoqingsaidemoyangzhongwendetizhuyis
houbazhishanfeileetcodeasuanfayoumeiyouxuejiushinagehenduo nantizuoliangbaidaojiunengjingugedeleetcodeming
baimaicpcfinalistmingbaileninjixushuotizhutafengkuangdichutinewoshuowohencaishiwenshuibuzhidaoranhourang
wozibijiushangzhouliunachangquanbudoushinantihaiyouxianduanshuxianduanshushujufanweinamedazhijierangwozuo
wonaqijianpanzhijiedalegebaolikuajiutlewojiuxiangrennoiaupuchitizhunixiaoshenmenoiauwoxiangqigaoxingdeshi
qingtizhushenmegaoxingdeshiqingnoiauwodeguangyihouzhuizidongjiparentshushangdongtaidianfenzhiacleicpcfina
listpuchitizhuniyouxiaoshenmeicpcfinalistewodeaczidongjishanggaosixiaoyuanjiegailvdpfangchengyeacletizhun
imenliangguodeshitongyigetinoiauduiduiduiicpcfinalistjiannandibiezhuxiaoabushishidouyongleshujujiegoutizh
uwozaizhongshenyibianwomeizaikaiwanxiaonoiauduiduinoiauiicpcfinalistmeirenzhuxiaochuleshengtizhuwaifengkua
ngqiaozhuoicpcfinalistwomenyanguizhengchuanninganggangshuodezhegeleetcodezhousaitananmetizhutabushinanbun
andewentitazhendeshinazhonghenshaojiandenazhongtadetimuhenqiaomiaodongtaiguihuaerfendaanhenyouyisiran
grenbaisibudeqijieyihandeshinatiantitaiduomeinengkanwannoaubiexiaotizhuniqirentaishenworennihenjiulenoiauwod
etiacletizhunimingmingzaixiaowonidoumeitingguonoiautizhuxianshengwomenshouguohenyangedexunlianwulunduohao
xiaonewomendoubuhuixiaochufeiren buzhu zhuantoukanxiangicpcfinalisticpcfinalistburuzheyangtizhuxianshengnin
xianhuiqu dengxiaoxiwomenyiyoujinzhanlimatongzhinintizhu hangnimenganjinquzuohennandeduodaidianrenjifangkua
ngxiaonoiautizhuxianshengninhaiyoushenmeyaobuchongmajifangkuangxiaoicpcfinalisttizhuxiansheng

输入

无

输出

你的答案

C. 异世界的融资(10分)

1 秒, 128 MB

三年之期已到。果冻花费三年时间学习了文字等知识，完全适应了异世界。他没有放弃自己开发游戏的梦想，于是四处奔波寻找若干王室融资，以筹备足够的资金启动项目。然而，王室间勾心斗角，每家王室都与多家王室直接敌对 (注：敌对是相互的)。如果他想得到某家王室的融资，就一定不能在与这家王室直接敌对的其他王室那里获得任何融资。果冻想要获得最多的融资，给定果冻打探到的王室敌对关系表，请帮他找到最佳融资方案，并输出最大融资资金数。

例如，假设敌对关系表如下所示：

王室编号	融资金额	直接敌对王室编号
1	114514	
2	1919180	3 4
3	1437580	2
4	1234567	2

那么可以向王室 1, 3, 4 寻求融资，最大融资资金数为 2786661。

对下列王室敌对关系表，求最大融资资金数。

1	76730720	4	6	13	22																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														</
---	----------	---	---	----	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

输入

无

输出

你的答案

D. 异世界的地图(10分)

1 秒, 128 MB

果冻有一份世界地图，上面标注了各魔法传送点及传送点两两间的传送路线，有一些路线是双向的、另外一些是单向的。如果传送路线是双向的，果冻可以从该路线的两点中任一点传送到另一点；如果传送路线是单向的，果冻只能从起点传送到终点。

由于极其特殊的气候，当前季节很长时间内恒定会出现一天雨天、一天晴天的循环交替。每次传送都需要消耗魔法。晴雨不同会影响魔法流，因此晴雨不同时，传送消耗的魔法也可能不同。

果冻听闻在某个传送点附近居住着名为禾枫的贤者，于是他打算前往拜访贤者。果冻当前在传送点 1，当天是晴天。从当天开始(含当天)，果冻每天都可以使用至多一次传送。由于果冻的魔法量很低，他想要在到达目的地时总消耗最小，请算出最小消耗。

例如，假设有传送点间情况如下表，想前往传送点 5，最小消耗是 19。

起点	终点	是否双向	晴天消耗	雨天消耗
1	3	Y	10	20
3	5	Y	10	22
2	1	N	1	2
3	2	N	10	2
2	4	Y	4	8
3	4	Y	30	32
1	4	Y	34	30
4	5	Y	15	3

假设有 $n = 1437$ 个传送点， $m = 4096$ 条传送路线。第 i 条传送路线的起点 u_i 是 $((1399i^2 + u_{i-1}^2) \bmod n) + 1$ ，终点 v_i 是 $((1427i^2 + v_{i-1}^2) \bmod n) + 1$ ，特别地，当计算得 $u_i = v_i$ 时，令 $u_i = (u_i \bmod m) + 1$ 。若 i 为奇数是双向传送路线；否则是单向传送路线。晴天消耗 x_i 为 i 的正因数之和加 x_{i-1} 后对 m 取模后再加一，雨天消耗 y_i 为 i 的不大于 m 的所有正倍数之和加 y_{i-1} 后对 m 取模后再加一。其中 $u_0 = v_0 = x_0 = y_0 = 0$ ，注意 1 是所有数的因数；所有数的因数和倍数包含其自身。求到传送点 n 的最小消耗。

假设你的程序生成的数据前五条传送路径如下所示，那么你的数据生成可能是正确的：

起点	终点	是否双向	晴天消耗	雨天消耗
1400	1428	Y	2	2049
1218	42	N	6	2
199	238	Y	11	3416
195	442	N	19	1369
1151	1120	Y	26	960

输入

无

输出

你的答案

E. 异世界的招聘(15分)

1 秒, 128 MB

果冻拜访了贤者禾枫，发现她也是穿越者，并且是自己的老相识。二人一拍即合，决定一同创业。贤者曾做过猎头，有很强的看人能力。已知，每个人有 1% 的概率是魔法师， 4% 的概率是其他职业者， 95% 的概率不符合条件。经决定，贤者打算为公司招聘 10 个魔法师， 80 个其他职业者。如果某个人符合条件且该职业未满足，则贤者立刻与之签订合同。贤者打算一直找人直到招满为止。设贤者一共要找 x 人才能招满，求 x 的数学期望 $E(x)$ ， 答案四舍五入保留小数点后六位。

例如，若招聘 1 个魔法师， 1 个其他职业者，数学期望为 $E(x) = 105.000000$ 。

输入

无

输出

你的答案

F. 异世界的剧情分支(15分)

1 秒, 128 MB

果冻发现贤者禾枫有文学天赋，于是让她构造剧情线。对于某一章节，贤者一共想了 m 个有区别的剧情分支和 n 个无区别的剧情背景。现要求每个剧情分支必须选择至少一个剧情背景作为基础，且要求每个背景必须被唯一的剧情分支选择。求有多少种剧情背景的选择方案。

输入

首先输入一行一个整数 t ，代表询问的个数。

接下来输入 t 行，每行两个整数，代表 n, m ，用一个空格隔开。

对 20% 分数的数据， $1 \leq n, m \leq 8$

对 40% 分数的数据， $1 \leq n, m \leq 10^3$

对 80% 分数的数据， $1 \leq n, m \leq 9 \times 10^4$

对 100% 分数的数据， $1 \leq n, m \leq 10^{18}, m \leq n, 1 \leq t \leq 10^5$

输出

输出 t 行，每行一个整数，代表每个询问的选择方案数对质数 99991 取模后的结果。

样例

样例输入 1	样例输出 1
3 5 5 5 2 5 3	1 4 6

说明

对 $n = m = 5$ ，只有一种方案：每个分支选一个背景

对 $n = 5, m = 2$ ，可以让两个分支分别选 $(1, 4), (2, 3), (3, 2), (4, 1)$ 个背景，共四种方案

对 $n = 5, m = 3$ ，可以让三个分支分别选 $(1, 1, 3), (1, 3, 1), (3, 1, 1), (1, 2, 2), (2, 1, 2), (2, 2, 1)$ 个背景，共六种方案

G. 异世界的人偶训练(20分)

1 秒, 128 MB

在果冻的指示下，贤者禾枫创造了大量魔法人偶辅助魔法师工作。为了训练魔法人偶的思考能力，她打算让人偶两两之间进行一种游戏，具体规则如下：

给定一个只包含 0 和 1 的字符串 s ，其长度为 n ，两个人偶轮流在以下两种操作中选择一种执行：

- 1. 将字符串反转，即从后往前重新书写字符串。如 00001 反转后变为 10000。这个操作不耗费魔力。
- 2. 将字符串中的 1 个 0 改为 1。耗费 1 点魔力。

当字符串中的每个字符都变为 1 时游戏结束。此时，耗费魔力更少的那个人偶获得胜利。若两个人偶所耗费的魔力相同，则游戏平局。

为了让游戏更加有趣，禾枫添加了一个限制：

操作 1 只能在当前字符串不是回文串，且前一次操作不是操作 1 时执行。

其中，回文串是指从后往前读和从前往后读都相同的字符串，即：记下标从 1 开始，长为 n 的回文串 s 满足对任意 $i (1 \leq i \leq n)$ ， $s[i] = s[n - i + 1]$ 。

在先手的第一次操作的情况下，不存在“前一次操作”，此时只需要满足字符串不是回文串的条件即可执行操作 1。

举个例子，当第一个人偶执行了操作 1 后，即使此时的字符串不是回文串，按照限制，第二个人偶仍然无法执行操作 1。反之亦然。

现在已知一个字符串，并且两个人偶都会以最优策略进行游戏，禾枫想要知道游戏的结果。

输入

输入一个整数 $T (1 \leq T \leq 10^3)$ ，代表接下来有 T 个测试用例。

对于每个测试用例：

输入一行一个整数 n 。

接下来一行输入一个长度为 n ，并且只包含 0 和 1 的字符串。

对于 20% 分数的数据， $1 \leq n \leq 5$ ， $\sum n \leq 10$ 且保证输入的字符串是回文串。

对于 60% 分数的数据， $1 \leq n \leq 10^3$ ， $\sum n \leq 10^3$ 且保证输入的字符串是回文串。

对于 100% 分数的数据， $1 \leq n \leq 10^3$ ， $\sum n \leq 10^5$ 且保证输入的字符串中至少有 1 个 0

输出

对于每个测试用例，第一行输出一个整数。若先手胜利，输出 1；若后手胜利，输出 -1；若游戏平局，输出 0

样例

样例输入 1	样例输出 1
2 1 0 2 00	-1 -1

说明

对于样例 1 中的测试用例 1：

字符串已经是回文串，先手只能执行操作 2，之后字符串变为 1，游戏结束。

此时先手耗费 1 点魔力，后手耗费 0 点魔力，后手获得胜利。

对于样例 1 中的测试用例 2：

字符串已经是回文串，先手只能执行操作 2，然后后手会执行操作 1，由于限制的存在，先手只能继续执行操作 2，游戏结束。

此时先手耗费 2 点魔力，后手耗费 0 点魔力，后手获得胜利。

H. 异世界的稳定测试(20分)

1 秒, 128 MB

果冻凭借着自己曾经所学的计算机组成原理，先是成功用魔法创造出了逻辑门，继而又创造出了魔法运作的计算机。为了机器稳定运转，魔法师需要为其施加尽可能稳定的魔法流。贤者禾枫认为，魔法流的标准差可以判定稳定与否。

有 n 个魔法师，编号从 1 到 n 递增，第 i 个魔法师能提供大小为 x_i 的魔法流。一开始他们按从 1 到 n 的顺序从左到右排成一队。左边是队首，右边是队尾。

有两种操作：

- 1. 移动：将队首前 k 个魔法师从左到右依次取出，按取出顺序逐个放到队尾。
- 2. 测试：让当前队伍从左往右第 l 到第 r 个(含第 l, r 个)魔法师输出魔法流，统计输出的标准差 s 。

贤者会进行多次上述操作，每次操作基于前一次的操作。

设平均数为 \bar{x} ，标准差 s 计算公式为：

$$\bar{x} = \frac{\sum_{i=l}^r x_i}{r-l+1}, \quad s = \sqrt{\frac{\sum_{i=l}^r (x_i - \bar{x})^2}{r-l+1}}$$

输入

首先输入一行两个整数 n, m

接下来输入一行 n 个整数，第 i 个整数代表 x_i

接下来输入 m 行，对于第 i 行，若输入两个整数 1 k ，代表进行移动操作；若输入三个整数 2 l r ，代表进行测试操作。

对于 20% 分数的数据， $1 \leq n \leq 10^3, 1 \leq m \leq 10^4$

对于 100% 分数的数据， $1 \leq n, m \leq 10^5, 0 \leq x_i \leq 10^6, 1 \leq k \leq n, 1 \leq l \leq r \leq n$

输出

对于每个测试操作，输出一行一个小数 s ，代表这次测试的能力值标准差。你的答案被认为是正确的当且仅当你的答案与标准答案的相对误差不超过 10^{-6} 。

样例

样例输入 1	样例输出 1
5 6 1 2 3 4 5 2 2 4 2 2 5 1 2 2 3 4 1 4 2 2 5	0.816497 1.118034 2.000000 1.479020

说明

对第一个测试，

$$\bar{x} = \frac{2+3+4}{3} = 3, \quad s = \sqrt{\frac{(2-3)^2 + (3-3)^2 + (4-3)^2}{3}} \approx 0.816497$$

对第二个测试，

$$\bar{x} = \frac{2+3+4+5}{4} = 3.5,$$

$$s = \sqrt{\frac{(2 - 3.5)^2 + (3 - 3.5)^2 + (4 - 3.5)^2 + (5 - 3.5)^2}{4}} \approx 1.118034$$

对第一次移动，移动后从左到右魔法师能力值为3, 4, 5, 1, 2

对第三个测试，

$$\overline{x} = \frac{5 + 1}{2} = 3.5, s = \sqrt{\frac{(5 - 3)^2 + (1 - 3)^2}{2}} = 2$$

对第二次移动，移动后从左到右魔法师能力值为2, 3, 4, 5, 1

对第四个测试，

$$\overline{x} = \frac{3 + 4 + 5 + 1}{4} = 3.25,$$

$$s = \sqrt{\frac{(3 - 3.25)^2 + (4 - 3.25)^2 + (5 - 3.25)^2 + (1 - 3.25)^2}{4}} \approx 1.479020$$

I. 异世界的模块合并(25分)

1 秒, 128 MB

果冻利用训练好的人偶和员工，成功制作了异世界的第一款二次元氪金卡牌手游，给异世界带来了巨大的文化冲击，一时成为现象级爆款。果冻并未满足，在贤者禾枫的建议下，果冻打算制作一款包含原世界所有内容的大型游戏，起名为果核大陆。

由于开发周期过长，果冻打算使用增量发布。最初发布的版本只保留最核心的玩法，其余玩法在后续版本加入。考虑到模块耦合度，最开始有 n 组模块，第 i 个模块单独在第 i 组，价值为 2^{n^2+i} 。模块和组编号均从 1 到 n 连续。

现在有 $n - 1$ 种合并方案，第 k 种合并方案有 k 次合并。在第 j 次合并，首先找到编号为 $((j + k) \bmod n) + 1$ 的模块所在组和编号为 $((j + k)^2 \bmod n) + 1$ 的模块所在组。若它们是同一组，将选择其中一个组更改为该组的下一个组(若存在编号更大的组，则是首个编号比它大的组；若不存在，则是当前编号最小的组)，然后合并这两个组为编号更大的组。合并时，先把编号较小组的所有模块放入编号较大的组，然后删除编号较小组。合并后编号较大的组价值不变。

为尽可能多合并，果冻想找到最大整数的 k ，满足合并后剩余各组价值之积大于 $(p!)^p$ 。

输入

输入一行两个整数 n, p 。

对 20% 分数的数据， $1 \leq n \leq 6, 0 \leq p \leq 6$

对 40% 分数的数据， $1 \leq n \leq 6 \times 10^3, 0 \leq p \leq 1.5 \times 10^5$

对 80% 分数的数据， $1 \leq n \leq 10^5, 0 \leq p \leq 10^5$

对 100% 分数的数据， $1 \leq n \leq 5 \times 10^5, 0 \leq p \leq 5 \times 10^5$

输出

若能找到这样的 k ，输出一行一个整数 k ；否则，输出 `impossible`。

样例

样例输入 1	样例输出 1
6 6	4

样例输入 2	样例输出 2
6 1	5

样例输入 3	样例输出 3
6 10	impossible

说明

对 $n = 6$ ，有 5 种方案，依次剩下的组编号为 $(1, 2, 4, 5, 6), (1, 2, 3, 6), (3, 4, 6), (4, 6), (6)$ 。

J. 异世界的魔法优化(25分)

2 秒, 128 MB

为制作果核大陆，果冻需要编织一个精妙的法术集合。一个法术集合由许多子魔法构成。经研究，贤者禾枫发现可以通过调整子魔法的消耗改变果核大陆的各种属性。于是他们开始研究如何对异世界现有魔法进行调整。

果冻发现，法术集合能以树的形式表示，树上每个节点代表组成该法术集合的子魔法，每个节点都要消耗能量。树有 n 个节点，编号从 1 到 n ，第 i 个节点的消耗为 a_i 。对于树上两点，定义这两点形成的链为这两点之间的最短路径。

为了调整果核大陆的各种属性并实时观测效果，果冻开始进行实验。果冻每次可以选择两个节点，对于这两个节点形成的链，果冻可以在以下 3 种操作中选择一种执行：

操作 1：减小链上的每个节点的消耗，即：对链上的每个节点 x ，其消耗减小 $a_x \& (-a_x)$ ，其中 $\&$ 是按位与运算。

操作 2：增大链上的每个节点的消耗，即：对链上的每个节点 x ，若 $a_x \neq 0$ ，则其消耗增大 2^k ，其中 k 是满足 $2^k \leq a_x < 2^{k+1}$ 的整数。

操作 3：查询链上所有节点的消耗之和对 998244353 取模的结果。

输入

输入一行一个整数 n

接下来输入一行 n 个整数，第 i 个整数代表初始消耗 a_i

接下来输入 $n - 1$ 行，每行两个整数 u, v ，代表节点 u, v 间连有一条边

接下来输入一行一个整数 q ，代表事件个数

接下来输入 q 行，每行三个整数 c, l, r

- 若 $c = 1$ ，对编号为 l, r 两点形成的链执行操作 1
- 若 $c = 2$ ，对编号为 l, r 两点形成的链执行操作 2
- 若 $c = 3$ ，对编号为 l, r 两点形成的链执行操作 3

对 20% 分数的数据， $1 \leq n, q \leq 10^3$

对 100% 分数的数据， $1 \leq n, q \leq 10^5, 1 \leq u, v, l, r \leq n, 1 \leq a_i \leq 10^9, 1 \leq c \leq 3$ ，保证输入的边能够形成一棵树

输出

对于每个查询，输出一行一个整数，代表编号为 l, r 的两点形成的链上的消耗之和对 998244353 取模的结果。

样例

样例输入 1	样例输出 1
5 5 2 2 9 7 1 2 1 3 3 4 3 5 4 1 2 5 3 1 5 2 5 4 3 4 2	10 21