

北京理工大学第九届程序设计新生赛

The 9-th BIT Campus Programming Contest for Junior Grade Group

现场赛 Onsite Round



题目列表 Problem List

A	出题人的碎碎念
B	页面置换
C	Rekop Medloh Saxet
D	法力风暴
E	肉夹馍
F	狂乱
G	理财大师
H	守门员
I	创业园的共享单车
J	zyw 范数
K	Easy Sigma
L	零时困境
M	客观的校长

请勿在比赛开始前翻阅试题!
Do not open before the contest has started.

2021 年 5 月 30 日

Problem A. 出题人的碎碎念

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

欢迎各位同学来到本届北京理工大学程序设计新生赛，作为本场比赛的出题人，也是 ACM 俱乐部的成员之一，我将为大家简要介绍本次比赛以及北京理工大学 ACM 俱乐部。

北京理工大学 ACM 俱乐部创立于 2018 年暑期，由此前的北京理工大学 ACM/ICPC 集训队和计算机学院学生科技创新创业基地算法实验室联合形成。团队创立以后，俱乐部成员齐手并进，取长补短，共同训练，为本年度的比赛积极地筹备，为获奖奠定了良好的基础。

ACM 俱乐部主要负责组织学生参加各种算法竞赛、程序设计竞赛，涉及到的比赛包括但不限于蓝桥杯、团体程序设计天梯赛 (CCCC)、中国大学生程序设计竞赛 (CCPC)、国际大学生程序设计竞赛 (ICPC)。之中，CCPC 与 ICPC 的各种区域赛是 ACM 俱乐部重点关注的赛事，该赛事设立金奖、银奖、铜奖。为了便于理解，2020 至 2021 年的大部分 ICPC 区域赛的奖牌发放规则可以粗略地概括为：金奖发放参赛队伍的 10%（不为整数则向下取整）、银奖（不含金奖人数）发放参赛队伍的 20%（不为整数则向下取整）、铜奖（不含金奖、银奖人数）发放参赛队伍的 30%（不为整数则向下取整）。

而在俱乐部成员的努力训练下，ACM 俱乐部成绩斐然，在最近于 2021 年 5 月 16 日举办的 ICPC 银川区域赛中，全部由 20 级新生组成的 ddl 战神队在 499 支参赛队伍中获得了冠军，是北理工历史上第一个区域赛冠军，也使得北理工时隔多年再次重返 ICPC 世界总决赛。此外，由 18 级同学组成的“迪杰斯特拉大战特斯拉”队、由 16 与 17 级同学组成的“putchar”队和由两名 18 级同学与一名 19 级同学组成“摸鱼两年半”队在本场比赛中分别获得了第 112 名、126 名、148 名的名次。

说回本次新生赛，本次新生赛面向所有大一和大二对程序设计竞赛有兴趣的同学，题目难度由易到难都有包含，但注意，题目顺序并不是按难度排序的，各位同学如果遇到不会做的题，可以先看看别的题目，出题人认为本场比赛可以上手做的题目还是比较多的，欢迎大家多多尝试。

本场比赛的出题人主要是“克鲁斯卡尔大战克鲁鲁”队的三位队员，该队在本年度的区域赛中一共获得过两块区域赛金牌与一块 **EC-Final** 铜牌，他们在算法竞赛网站 Codeforces 上的用户名分别是“bit_liyiwen”、“chenhongta0”、“zhanggengchen”（不含引号）。此外，出题人还有用户名为“strawberry”的龙神、用户名为“shenyunhan”的沈帝，以及用户名为“fried-chicken”的 zyw。事实上，这次比赛的许多题目当中，都可以看到 zyw 这个名字的出现，但一个有趣的事实是：所有题目里出现了 zyw 的题目，反而都不是 zyw 出的，是其他出题人为了搞 zyw 而出的，而所有题面里没有出现 zyw 的题目，反而全部是 zyw 出的，唯一的例外是本题，本题题面中虽然出现了 zyw，但也是 zyw 出的。

最后，也感谢睿信科协、各位学长、志愿者、辅导员、老师与校领导提供的各种帮助、技术支持与物资支持，提前预祝本届程序设计新生赛圆满成功！

阅读完上面这段文字后，请你回答以下五个问题：

问题一：请输出三个整数，依次表示北京理工大学在 2021 年 ICPC 银川区域赛中获得的金牌、银牌、铜牌数量。

问题二：本场比赛中，出题人为“fried-chicken”的题目有几道？输出一个整数表示你的答案。

问题三：根据文章信息，参加了 2021 年 ICPC 银川区域赛的同学中，最多有几位同学可能参与本次程序设计新生赛？输出一个整数表示你的答案。

问题四：“克鲁斯卡尔大战克鲁苏”的三位成员的 Codeforces 用户名中，一共出现了多少个非小写英文字母的字符？输出一个整数表示你的答案。

问题五：下列四个说法中，有几个是正确的？输出一个整数表示你的答案。

说法一：2018 年的暑期之前，北京理工大学并没有负责组织参与 ICPC 的组织；

说法二：小明在本次比赛中发现 C 题不会做，因为 C 题后面的题肯定都比 C 题难，所以小明应该放弃本次比赛，回宿舍睡大觉；

说法三：本次新生赛的成功举办，全部都是 ACM 俱乐部的功劳；

说法四：ddl 战神队是北理工历史上第一个进入 ICPC 世界总决赛的队伍。

上述五个问题的答案可以用七个整数表示（问题一含三个整数，其余四个问题每题一个整数）。请你输出一行七个整数，表示这五个问题的答案。

特别地，由于出题人心地善良，允许你回答的五个问题中有一个问题是错误的，即，**你回答对五个问题或四个问题**，都将收获一个绿绿的 *correct*。

Input

本题没有输入，你只需要输出答案。

Output

输出七个空格分隔的整数，表示五个问题的答案。五个问题中答对四个即可通过此题。

Example

standard input	standard output
(No input)	1 2 3 4 5 6 7

Note

上方的样例只用于描述题目输入输出的格式，不代表输出内容就是对应问题的正确答案。

上述输出的含义是，答题人认为：问题一的答案是 1,2,3；问题二的答案是 4；问题三的答案是 5；问题四的答案是 6，问题五的答案是 7。

Problem B. 页面置换

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在地址映射过程中，若在页面中发现所要访问的页面不在内存中，则产生缺页中断。当发生缺页中断时，如果操作系统内存中没有空闲页面，则操作系统必须在内存选择一个页面将其移出内存，以便为即将调入的页面让出空间。而用来选择淘汰哪一页的规则叫做页面置换算法。

通俗的理解：现代计算机常采用页式存储，也就是把各种内容切分成固定大小的块进行存储。操作系统在使用某个页面的内容时，只能从内存中读取内容，所以当要使用某个页面时，如果其不在内存中，就需要将其先从外存调入内存中，但是显然内存的大小是有限的，当新的页面将要进入内存时，如果内存已满，就需要淘汰掉现有的页面，所以需要一些策略来对页面进行调度，这种策略就被叫做页面调度算法。

缺页中断：当调用某页面时，若其不在内存中，就发生缺页中断。

下面给出三种调度算法

1.FIFO 调度算法当需要淘汰一个页面时，总是选择驻留主存时间最长的页面进行淘汰，即先进入主存的页面先淘汰。

2.LRU 调度算法当需要淘汰一个页面时，总是选择在最近一段时间内最久不用的页面予以淘汰。即淘汰最近最长时间未访问过的页面。

3.OPT 调度算法当需要淘汰一个页面时，从主存中移出永远不再需要的页面；如无这样的页面存在，则选择最长时间不需要访问的页面。

zyw 想知道这三种调度算法的优劣，他会以缺页次数为指标，所以请你告诉他哪一种方法缺页中断次数最少，并且告诉他会发生多少次缺页中断。（假定一开始内存中没有任何页面）

Input

第一行给定三个整数 n, m, k ，分别表示页面请求次数，外存中页面个数（从 1 开始编号），内存的大小。其中 $n \leq 10^5, m \leq 10^3, k \leq 100$ 。

第二行给出 n 个 1 到 m 之间的整数，表示页面调用请求的顺序。

Output

第一行输出最优的方法，如果有两种或更多方法是最优的，输出任意一种都可以。

第二行输出缺页中断的次数。

Example

standard input	standard output
12 8 4 6 2 4 1 4 6 3 8 4 5 7 3	FIFO 8

Problem C. Rekop Medloh Saxet

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

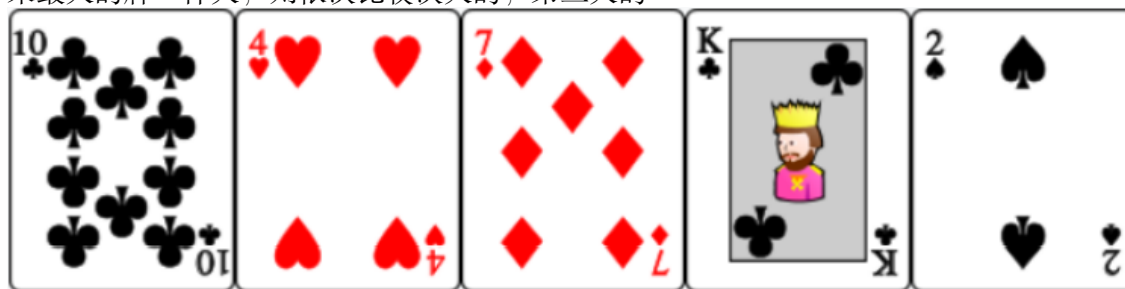
有一种奇怪的游戏，叫做 Rekop Medloh Saxet。它的游戏规则十分奇怪，请认真阅读以下的规则之后再做题。游戏中，每人有两张私有牌，这两张私有牌只能由所属的玩家查看。牌桌上有三张公共牌，着三张牌所有人可以看到。游戏使用一副扑克牌进行游玩（无大小王）。游戏所用的扑克牌和平时大家玩的扑克牌没有区别。一副标准的 52 张的扑克牌有四种花色，每种花色有十三张大小不同的牌。四种花色如下：Spade ♠ 黑心、Heart ♥ 红心、Diamond ♦ 方片、Club ♣ 草花，简写分别为 S,H,D,C。而每种花色都有 Ace(A),King(K),Queen(Q),Jack(J),10(T),9,8,7,6,5,4,3,2 这 13 张牌（从大到小）。此外，可以认为不同花色的同一张牌是一样大的，例如：♥K 和 ♣K 一样大。注意：每局游戏中同一张牌最多出现一次。

Example set of 52 playing cards; 13 of each suit: clubs, diamonds, hearts, and spades

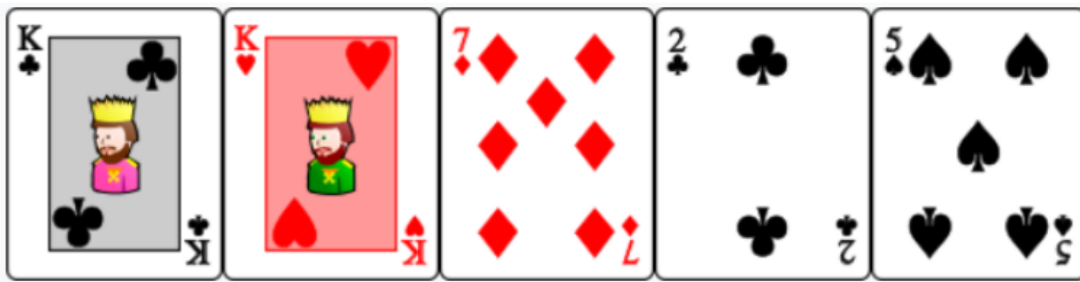
	Ace	2	3	4	5	6	7	8	9	10	Jack	Queen	King
Clubs													
Diamonds													
Hearts													
Spades													

游戏开始后每个人把自己的两张私有牌和桌上的三张公共牌组合到一起，进行比大小，大的人获得胜利。如果两人牌型不同，则比较牌型，牌型大的获胜；牌型相同则按每种牌型的规则进行比较。下面将按从小到大的顺序给出所有的牌型：

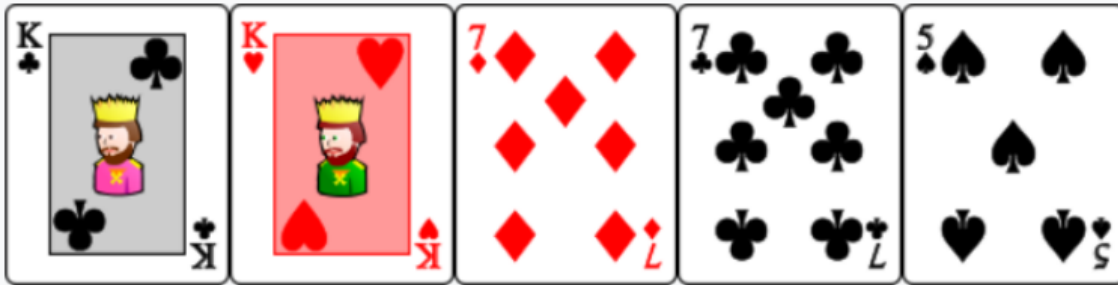
High Card/高牌：不满足之后所有的牌型的牌型，比较方式为从最大的牌开始比较，更大的人赢，如果最大的牌一样大，则依次比较次大的，第三大的……



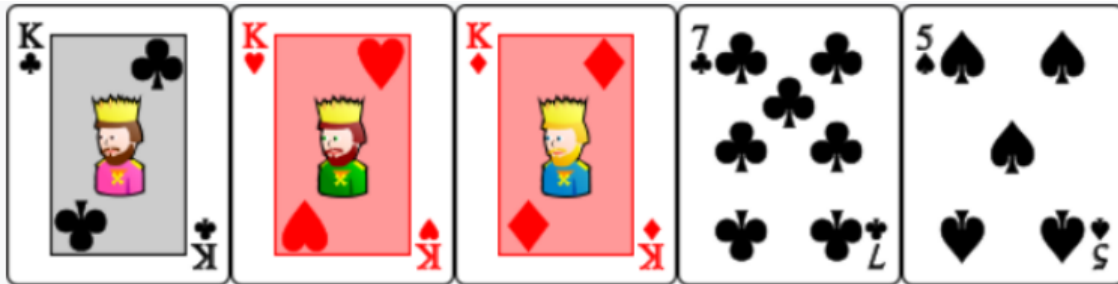
Pair/对子：有两张牌一样大（对子），其余三张牌和对子牌都不一样大且互不相同。比较方式为，先比较对子的大小，大的获胜；如果对子一样大，其余三张牌同 High Card 的方法比较。



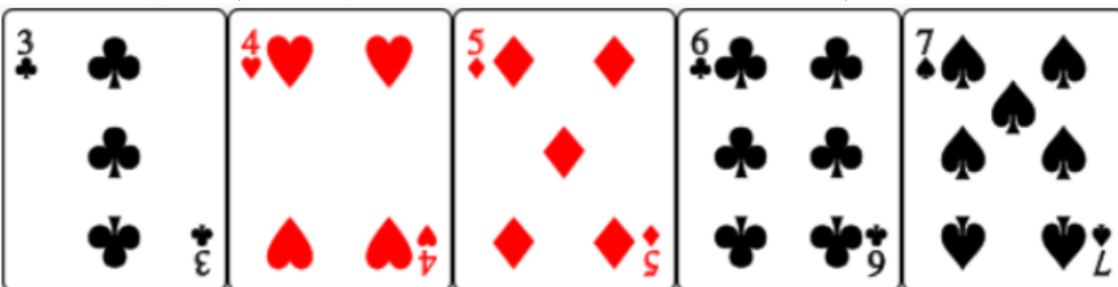
Two Pairs/两对：有两个不同大小的对子，第五张单牌和两个对子大小不同。比较方法为先比较较大的对子，大的获胜；一样大则比较次大的对子，大的获胜；否则比较第五张牌，大的获胜。



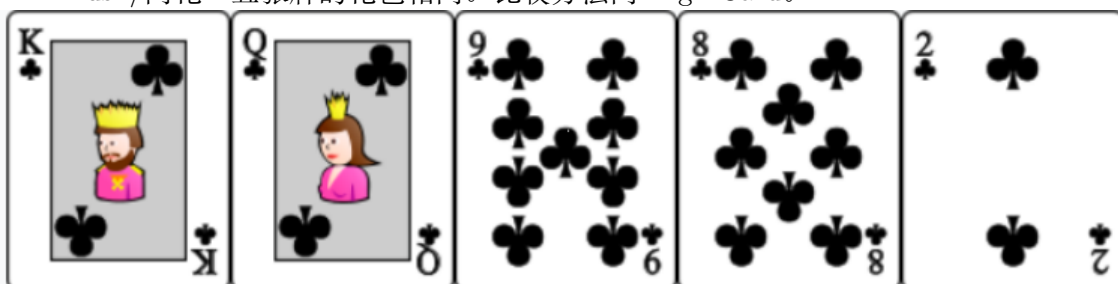
Three of a Kind/三条：有三张牌一样大（三条），其余两张牌和三条的牌都不一样大且互不相同。比较方式为，先比较三条的大小，大的获胜；如果三条一样大，其余两张牌同 High Card 的方法比较。



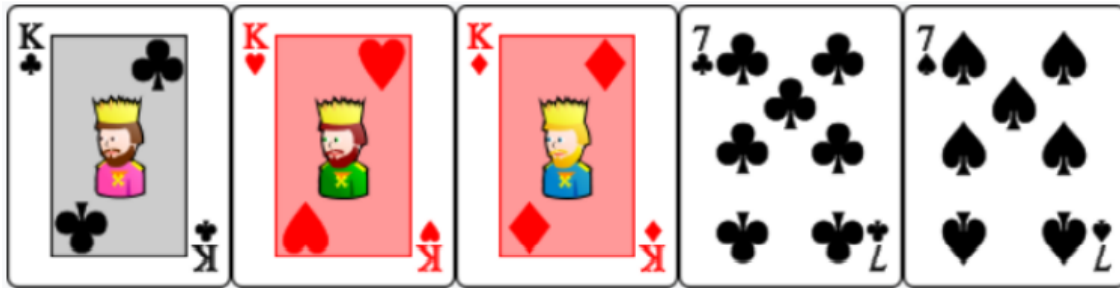
Straight/顺子：五张牌从大到小排序后相邻两张总是差一个等级，就叫做顺子。比较方法为比较最大的一张牌。特别地，可以认为 A2345 也是顺子但是它比别的顺子都要小。



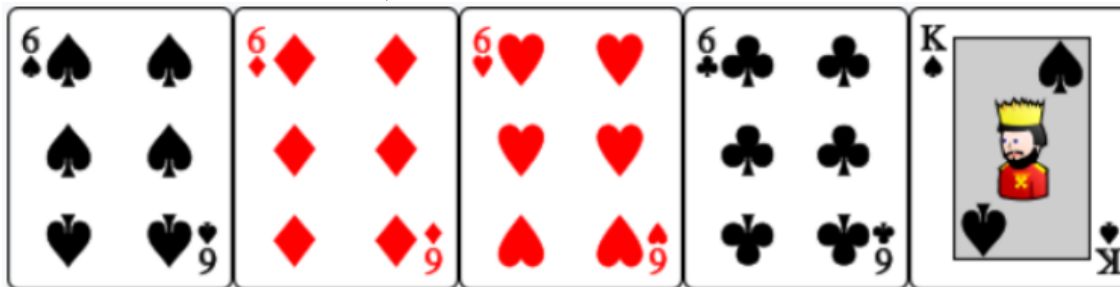
Flush/同花：五张牌的花色相同。比较方法同 High Card。



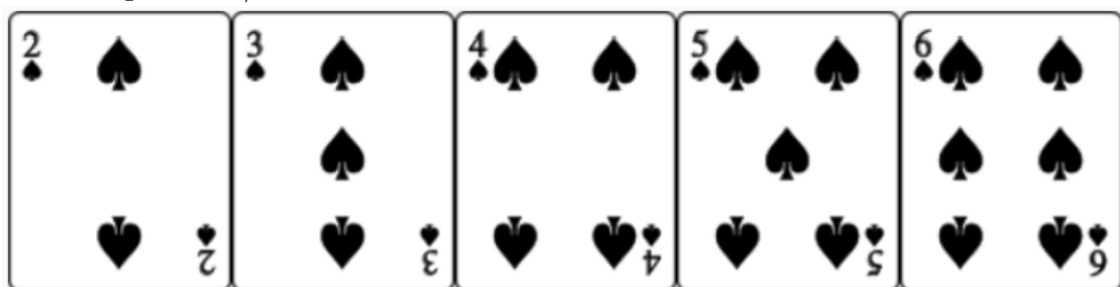
Fullhouse/葫芦：有大小不同的一个对子和一个三条。比较方法为先比较三条，大的获胜；相同则比较对子，大的获胜。



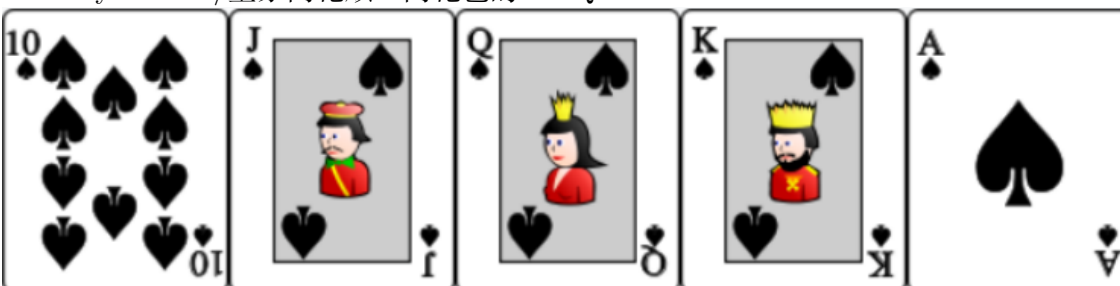
Four of a Kind/四条：有四张大小相同的牌和一张不一样的其他牌。比较方法为先比较四条，大的获胜；相同则比较剩下的一张牌，大的获胜。



Straight Flush/同花顺：是顺子且是同花。比较方法同顺子。



Royal Flush/皇家同花顺：同花色的 AKQJT。



你正在和 zyw 玩 Rekop Medloh Saxet，你现在已经看到了手里的牌和公共牌。鉴于 zyw 的强大实力（不清楚是哪方面的），你想知道这局能不能 100% 赢。所以只有一定能赢时你才会继续游戏（all in）。否则你只能弃牌（check）。

Input

第一行给出 $T(T \leq 10^5)$ 。

之后 T 行每行给出五张牌。

前两张是你手里的牌，后三张是桌上的牌。

Output

对于每行的五张牌，输出两行：

第一行输出是什么牌型（英文）。

第二行输出是否一定能赢，一定能赢输出 allin，否则输出 check。（输出小写字母）。

Examples

standard input	standard output
1 SA SK SQ SJ ST	Royal Flush allin
11 SA SK SQ SJ ST S2 S3 S4 S5 S6 SA SK SJ ST S9 S7 H7 S3 D5 D7 S7 H7 S2 S3 D7 S7 H7 D7 C7 H3 H7 H3 D7 S7 D3 H7 H3 D3 D7 D2 H7 D7 C3 C4 D5 H3 D4 C5 H6 D7 SA S3 H5 H7 D9	Royal Flush allin Straight Flush check Flush check Three of a Kind check Three of a Kind allin Four of a Kind allin Fullhouse check Two Pairs check Pair check Straight check High Card check

Problem D. 法力风暴

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

zyw 最近迷上了一款 TCG(Trading Card Game) 桌游, 万智牌。在万智牌中有着 5 种不同的基础法力。玩家可以使用不同的法力来释放不同的咒语来进行史诗级的对决。

为了简化题目, 我们假定只有红色与蓝色的法力。

在游戏的某个回合中, zyw 拥有着 n 张可以释放的咒语, 每个咒语都有其法力的消耗, 第 i 个咒语的消耗是 r_i 个红色法力和 b_i 个蓝色法力。

zyw 突然发现自己竟一个法术都无法释放, 他十分无奈只能跳过这一回合。

请问 zyw 这回合最多可能拥有多少点法力 (红色和蓝色的总和)。

Input

第一行给出一个整数 T , 表示题目一共有 T 组数据。

对于每组数据, 第一行给出一个整数 $n(n \leq 5000)$ 。

第 2 到 $n+1$ 行每行给出两个整数, 第 $i+1$ 行的两个整数表示 r_i 与 $b_i(r_i \leq 10^6, b_i \leq 10^6)$ 。

保证对于任意的 i , 有 $r_i \geq 0, b_i \geq 0, r_i + b_i > 0$ 。

保证 $\sum n \leq 5000$

Output

对于每组数据, 输出一个整数, 表示 zyw 最多可能拥有的法力点数, 如果是无限点, 则输出 *INF* 三个大写字母。

Example

standard input	standard output
2	3
3	INF
0 3	
3 0	
2 2	
3	
1 3	
2 2	
3 1	

Problem E. 肉夹馍

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

zyw 是一个旅行爱好者，同时也是一个吃货。zyw 最喜欢的事情是到各地旅游并品尝当地的美食。这一次他来到了陕西西安旅游。众所周知，肉夹馍是陕西西安的传统小吃，有着馍香肉酥，肥而不腻，回味无穷的特点。zyw 品尝了肉夹馍之后开始思考，为什么没有“肉夹馍夹馍”呢？又为什么没有“馍夹肉夹肉”呢？

于是，他想让卖肉夹馍的师傅为他定制一个肉夹馍，他用一个 01 串表示他的理想肉夹馍，其中 0 表示肉，1 表示馍，字符串从左到右对应着肉夹馍的从里到外。

我们规定 ‘*’ 代表肉，‘-’ 代表上下两层的馍，‘|’ 代表右侧的馍片。同时规定，肉夹馍都是向左侧开口，并且最里层的肉或馍长度为 2 个相应字符，外层的肉或馍宽度都为 1 且紧紧包裹着里层的肉或馍。

由于题目过于抽象，可以参考样例来进行进一步的理解。

例如，对于样例五的串 011，对应的肉夹馍最里层是肉，外面包了两层馍。最里层的肉用两个 ‘*’ 表示，靠里的一层馍用肉周围的 7 个字符表示，靠外的一层膜用靠里的一层膜周围的 11 个字符来表示。

现在，请你将表示 zyw 定制的肉夹馍的 01 串转化为图形展示给卖肉夹馍的师傅吧！

Input

输入一行字符串，表示 zyw 的理想肉夹馍，串长在 1 到 1000 之间且保证只含有 0 和 1。

Output

输出 zyw 定制版肉夹馍的图形。

Examples

standard input	standard output
1	--
0	**
01	--- ** ---
10	*** --* ***
011	---- --- ** --- ----
001	---- *** *** *** ----

Problem F. 狂乱

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在游戏炉石传说中，有一张叫狂乱的法术牌，其效果是选择一个随从，使其随机攻击其他随从直至死亡。

现在 zyw 正在玩炉石传说，但是当前场上没有一个他的随从，而他的对手有 n 个随从，第 i ($1 \leq i \leq n$) 个随从有两个属性值，分别是攻击力 x_i 和血量 y_i 。随从在一次攻击时，攻击方和被攻击方都会减少对方攻击力的血量，如果攻击或被攻击后血量小于等于 0 则该随从死亡。

现在 zyw 想使用一张狂乱指定敌方的一个随从，他想知道最好情况下对方剩余随从的攻击力总和最小为多少。

Input

第一行一个整数 n ($1 \leq n \leq 100$) 接下来 n 行，每行两个整数 x, y ($1 \leq x_i, y_i \leq 10^4$)，表示每个敌方第 i 个随从的攻击力和血量

Output

一个整数，表示使用一张狂乱最好情况下敌方剩余随从的攻击力总和为多少。

Examples

standard input	standard output
4 1 1 1 3 2 2 3 5	3
6 1 3 5 4 3 5 1 1 1 1 1 1	1

Problem G. 理财大师

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

因为 *zyw* 穷的离谱，根本没有多余的钱拿来理财，所以本题的主角并不是 *zyw*。

最近学校食堂的价格越来越贵，原本就吃得多的懋懋这下要花更多钱用来吃饭了。为了避免食堂价格进一步增长，懋懋发行了一种全新的加密货币，北理币 (BIT coin, *bitc*)，用于学校食堂消费，来防止通货膨胀带来的食堂价格上涨。和其他加密货币一样，北理币和人民币的汇率 (1 *bitc* 能换多少人民币) 也会随各种因素而变化，不过作为北理币的发明人，懋懋可以预知北理币的汇率。

龙神作为懋懋的队友，在某次和懋懋聊天时得知了当天接下来 n 个时刻北理币的汇率，龙神肯定不会错过这个理财机会，但是在用人民币和北理币进行交易时，存在以下几个条件：

- 单笔最小交易量为 0.01 *bitc*，无单笔最大交易量限制；
- 由于龙神偏好谨慎投资，在任何时刻，龙神手里都不能持有超过 1 *bitc*；
- 每时刻可以进行任意多笔买卖交易；
- 每天最多可以进行 k 笔交易；

龙神初始不持有任何 *bitc*，但有无穷多的人民币，你能帮龙神算算他当天最多能赚多少人民币吗？

Input

第一行 2 个整数 n 和 k ($1 \leq n, k \leq 1000$)，分别表示当天的时刻数和每天最多交易次数。

第二行 n 个实数 $p_1, \dots, p_n$ ($0 \leq p_i \leq 100$)，表示 n 个时刻的北理币汇率，小数点后最多 2 位。

Output

一个实数，表示龙神能够获得的最大收益。

若你的输出为 a ，标准答案为 b ，则当 $\frac{|a-b|}{\max(1,b)} \leq 10^{-4}$ 时，就认为正确。

Examples

standard input	standard output
4 4 1 2 1.5 2.5	2.00
4 2 1 2 1.5 2.5	1.50

Note

对于样例 1，在第 1、3 时刻买入 1 *bitc*，在第 2、4 时刻卖出 1 *bitc*，总收益为 2*rm*b；

对于样例 2，在第 1 时刻买入 1 *bitc*，在第 4 时刻卖出 1 *bitc*，总收益为 1.5*rm*b。

Problem H. 守门员

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

zyw 是一名出色的足球前锋，人称“良乡内马尔”。他在紧张的学习生活之余也不忘记练习自己的足球技巧。这一天他在操场上练习射门，我们将球场看成一个二维平面，宽度为 $2 * w$ 的球门的两个门柱坐标分别为 $(-w, 0)$ 和 $(w, 0)$ 。

现在足球摆放在 (x_1, y_1) 的位置，zyw 要在此处射门。此时守门员的位置在 (x_2, y_2) ，守门员在 zyw 射门前能够移动的范围为 d ，并且只能水平移动。zyw 想立刻计算出他能够进球的角度有多少。

Input

第一行，一个数 $T (1 \leq T \leq 10^5)$ ，表示 zyw 射门的次数。

接下来 T 行，每行六个整数 w, x_1, y_1, x_2, y_2, d ，含义见题面描述，数据保证 y_1 和 y_2 不相等，数据范围 $0 < y_1 \leq 10^7$ ， $0 \leq y_2 \leq 10^7$ ， $-10^7 \leq x_1, x_2 \leq 10^7$ ， $0 < w, d \leq 10^7$ 。

Output

对于每一次射门，计算 zyw 射门能进球的角度，用弧度制表示。

若你的输出为 a ，标准答案为 b ，则当 $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ 时，就认为正确。

Example

standard input	standard output
3	0.7853981634
2 0 2 1 1 1	1.5707963268
2 0 2 100 1 5	0.0000000000
2 0 2 5 1 100	

Note

样例中三个答案分别对应， $\frac{\pi}{4}$, $\frac{\pi}{2}$, 0。

C 语言中反三角函数有 `acos`，`asin`，`atan`，返回值为弧度制。

需要 `#include<math.h>` 或 `#include<cmath>`

Problem I. 创业园的共享单车

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

ACM CLUB 的成员们经常在创业园开展训练, 为了节省宝贵的训练时间, 大家都喜欢骑共享单车来回。但遗憾的是, 每到晚上, 创业园门口的共享单车可能会供不应求。也就是说只有一部分人能扫上共享单车, 另一部分人只能步行。为了体现团队精神, 大家约定要同时到学校, 即到得早的人会等待到得晚的人。所以即使有一部分人扫上了共享单车, 到达学校的时间还是等同于步行。

为了让大家更快到达学校, 聪明的 zyw 想了个办法。以 2 人 1 车距离 100 米为例: A 先骑车至 50 米处, 下车开始步行, B 步行至 50 米处, 看到 A 放下的车并开始骑, 骑至 100 米处两人会合。这样一来, 到达时间显然短于全程步行的用时。

现在有 T 组信息, 每组信息包括人数 n , 共享单车数 m , 创业园到学校的距离 L 米, 大家的步行速度 v_p 米/秒, 大家的骑车速度 v_b 米/秒, 请你告诉 zyw 大家到达学校最少需要多少秒。为了简化问题, 所有人的步行速度和骑车速度视作相同且骑车速度大于步行速度, 上下车时间忽略不计, 随地停放的共享单车不会被路人骑走。

Input

第一行一个整数 $T(1 \leq T \leq 10^5)$

接下来 T 行, 每行五个整数 $n(1 \leq n \leq 10^9)$, $m(0 \leq m \leq 10^9)$, $L(1 \leq L \leq 10^9)$, $v_p(1 \leq v_p \leq 10^9)$, $v_b(v_p < v_b \leq 10^9)$, 含义如上所述。

Output

输出 T 行, 每行一个数表示到达学校的最短时间。

若你的输出为 a , 标准答案为 b , 则当 $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ 时, 就认为正确。

Example

standard input	standard output
1	75.0000000000
2 1 100 1 2	

Problem J. zyw 范数

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

zyw 是北京理工大学 ACM 俱乐部出了名的数学大师，没有哪道数学题可以难到他的。每次的比赛中如果有数学题那么他一定是第一个解出这题的人。作为北理数学界的泰斗，zyw 大佬也很平易近人，即使是面对数学小白他也会很耐心地为他们答疑解惑。

为了纪念 zyw 大佬为北京理工大学 ACM 俱乐部作出的贡献，我们设计并定义了矩阵的 zyw 算子和 zyw 范数。

- **zyw 算子**：对于任意的 n 阶整数方阵 A 和给定的整数 c ，定义作用在矩阵第 i ($1 \leq i \leq n$) 行第 j ($1 \leq j \leq n$) 列的 zyw 算子为将该位置元素 $a_{i,j}$ 替换为 c ，记作 $\text{zyw}_c(A, i, j) : \{A \mid a_{i,j} \leftarrow c\}$ 。
- **zyw 范数**：对于给定的 n 阶方阵 A 和整数 c ，定义 zyw 范数为使用 zyw 算子作用在 A 矩阵某行某列后得到的新矩阵 A' 中所有可能的新矩阵行列式的最大值，记作 $\text{zyw-norm}_c(A) = \max_{1 \leq i, j \leq n} |\text{zyw}_c(A, i, j)|$ 。

例如，给定 2 阶方阵 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 和整数 5，zyw 算子作用在矩阵 A 的第 1 行第 1 列后得到的新矩阵 $A'_{1,1} = \begin{bmatrix} 5 & 2 \\ 3 & 4 \end{bmatrix}$ ，其行列式 $|A'_{1,1}| = 14$ ，作用在矩阵 A 的第 1 行第 2 列后得到的新矩阵 $A'_{1,2} = \begin{bmatrix} 1 & 5 \\ 3 & 4 \end{bmatrix}$ ，其行列式 $|A'_{1,2}| = -11$ ，同理 $A'_{2,1} = \begin{bmatrix} 1 & 2 \\ 5 & 4 \end{bmatrix}$ ，其行列式 $|A'_{2,1}| = -6$ ， $A'_{2,2} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$ ，其行列式 $|A'_{2,2}| = -1$ ，故矩阵 A 的 zyw 范数则为矩阵 $A'_{1,1}$ 、 $A'_{1,2}$ 、 $A'_{2,1}$ 和 $A'_{2,2}$ 们行列式的最大值，即 $|A'_{1,1}| = 14$ 。

Input

第一行输入一个正整数 T ($1 \leq T \leq 1\,000$)，表示数据组数。

接下来 T 组数据，每组数据第一行输入两个整数 n ($1 \leq n \leq 3$) 和 c ($-100 \leq c \leq 100$)，由空格间隔开，分别表示输入矩阵的维度 n 和给定的整数 c 。

接下来 n 行，第 i ($1 \leq i \leq n$) 行输入 n 个整数 $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ($-100 \leq a_{i,j} \leq 100$)，由空格间隔开，分别表示矩阵第 i 行的元素。

Output

对于每组测试数据，请输出一个整数，表示该矩阵关于给定整数 c 的 zyw 范数，注意换行。

Example

standard input	standard output
1 2 5 1 2 3 4	14

Problem K. Easy Sigma

Input file: standard input
Output file: standard output
Time limit: 3 second
Memory limit: 256 megabytes

$$\sum_{i=1}^n (-1)^{\lfloor i\sqrt{k} \rfloor}$$

The value of $\lfloor x \rfloor$ is the biggest integer not exceed x .

Seeing this expression, zyw don't understand it, but greatly shocked. Therefore, he asked you to calculate the value of this expression.

Input

First line gives a integer T ($T \leq 10^5$)

The 2nd to T+1th lines, each line gives 2 integers n and k , meaning as the legend. ($n \leq 10^9, k \leq 10^4$)

Output

For each test case, print a integer, represents the answer.

Example

standard input	standard output
2	-1
3 1	1
3 2	

Note

It's easy to calculate the example

$$(-1)^{\lfloor 1\sqrt{1} \rfloor} + (-1)^{\lfloor 2\sqrt{1} \rfloor} + (-1)^{\lfloor 3\sqrt{1} \rfloor} = (-1)^1 + (-1)^2 + (-1)^3 = -1$$

$$(-1)^{\lfloor 1\sqrt{2} \rfloor} + (-1)^{\lfloor 2\sqrt{2} \rfloor} + (-1)^{\lfloor 3\sqrt{2} \rfloor} = (-1)^1 + (-1)^2 + (-1)^4 = 1$$

Problem L. 零时困境

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

当法伊、西格玛、戴安娜再次醒来时，发现他们又一次被关在了一个全新的密室中，有了之前几次的经验，他们显得并不吃惊。果然，密室里的小电视里，又出现了零的身影，他沙哑的嗓音也又一次从电视的扬声器中传出：“别担心，这次只要玩一个猜谜游戏，猜中了，你们就可以安全的离开这里”。

“零这家伙，一定又在捣什么鬼，我才不信他会有这么好心。”西格玛喃喃自语道。

游戏的规则是：零首先写下一个 n 个数的全排列 $\{p_n\}$ ，并告诉三人他选择的 n 。

接着，三人可以进行 m 次提问，每次提问，三个人每人给出一个数字，要求三个数字两两不相同，记为 i, j, k ，用于询问 p_i, p_j, p_k 中的中位数。接到提问后，零会如实回答三个数中中位数是哪一个，告知他们三人中位数的下标（即，零总是诚实地回答 i, j, k 三个数之一）。

现在，你知道了这 m 次提问的具体内容与零的回答，请你判断，三人是否可以通过这些信息 100% 的确定零手中的排列。

备注：一个含有 n 个数的全排列 $\{p_n\}$ 是指一个长为 n 且 1 到 n 的每个数都出现且仅出现一次的序列。例如， $[2, 1, 3], [3, 1, 2]$ 是全排列，而 $[1, 2, 4], [2, 3, 3]$ 都不是全排列。

Input

输入的第一行是两个整数 $n, m (1 \leq n \leq 10^5, 0 \leq m \leq 10^5)$ ，表示全排列的长度与三人提出的询问数。

接下来 m 行，每行四个数字 $i, j, k, ans (1 \leq i, j, k, ans \leq n, ans \in \{i, j, k\})$ 表示一次提问，含义如题面所述。

保证每次提问的 i, j, k 互不相同但有可能出现相同的提问、保证至少有一个长为 n 的全排列可以满足所有的 m 次询问的信息。

Output

请输出一行字符串，若三人可以唯一确定该全排列，输出 YES，否则输出 NO。

Example

standard input	standard output
4 2 1 2 3 2 4 1 3 4	NO

Note

对于样例， $[1, 2, 4, 3], [1, 3, 4, 2], [4, 2, 1, 3], [4, 3, 1, 2]$ 都是满足条件的全排列，因此三人无法确定全排列到底是什么。

Problem M. 客观的校长

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

我们常说,细节可以体现出一个人的情商高低,例如在北京理工大学,当同学们在教室(比如综 A105)或图书馆的长桌上自习时,如果有别的同学突然紧挨着坐在自己的旁边,往往就会令人感到尴尬和不适(偶尔还会有异味),我们称之为低情商行为。

现在,有一张含有 n 个座位的长桌,北理工同学认为,若一个学生在选择自习座位时,选择了一个周围(左侧或右侧) k 个座位(含 k 个)已经有人的座位,那么该同学就做出了一次低情商行为。

例如,若当前第 3 个座位已经有人,且 $k = 1$,则若此时有人来到第 2 个或第 4 个座位,则此人就是做出了低情商行为的。但原先在第 3 个座位的人并不被视为低情商的(因为他来到第 3 个座位时第 2 个和第 4 个座位并没有人)。

某一天,校长来到北理工视察北理工同学的情商水平,他看到了一张长桌 n 个座位的使用情况,但不知道现在教室中同学到来的先后顺序。由于校长分析问题是十分客观的,所以他想知道两个指标,一个是在最乐观的情况下,最少有多少人做出了低情商行为;一个是在最悲观的情况下,最多有多少人做出了低情商行为。

Input

第一行两个整数 $n, k (1 \leq n \leq 2 * 10^5, 1 \leq k \leq 100)$, 含义如题面所述。

第二行是一个长度为 n 的字符串,保证只含 0 与 1, 之中 0 表示当前座位为空, 1 表示当前座位有一位同学在坐着。

Output

输出一行, 包括两个整数, 依次表示最少的做出低情商行为的同学个数和最多的做出低情商行为的同学个数。

Examples

standard input	standard output
3 1 111	1 2
5 1 10101	0 0

Note

在第一个样例中, 用数字 1, 2, 3 表示当前使用这三个座位的人, 若到来顺序是 2 - 1 - 3, 则 1, 3 位置上的人做出了低情商行为, 若到来顺序是 3 - 1 - 2, 则只有 2 位置上的人做出了低情商行为。且枚举所有可能的情况之后, 可以发现最多只有 2 个同学可能做出低情商行为, 最少只有 1 个同学可能做出低情商行为。