

Problem [A]kali's Balance

time limit 2 seconds / memory limit 256 megabytes

Description

艾欧尼亚群岛上存在一个上古教派，致力于维护某种均衡。规则与混乱、光明与黑暗——万物必须和谐共存，这就是宇宙的真理。该教派名为均衡教派，维护世界均衡的神圣使命则由三个暗影战士来执行。阿卡丽便是这三名暗影勇士中一员，她肩负着修枝的神圣职责——将威胁瓦洛兰大陆和谐的敌人消灭干净。

阿卡丽自小和母亲一起习武，练就一身好武功。母亲的训练严酷无情，她的基本原则是：“我们是在替天行道”。阿卡丽十四岁加入均衡教派，那时她就能空手砍断链条。毋庸置疑，她将继承母亲“暗影之拳”的名号。她的所作所为别人可能无法理解，但在阿卡丽看来，她在执行母亲神圣的信条。她要 and 慎和凯南一起维护着瓦洛兰大陆的平衡。这一神圣的追求毫无疑问地将三个暗影忍者引领向正义之地。

作为一个捍卫均衡的忍者，她很早以前就被师傅嘱托需要监视另一块神秘的大陆 HUXI：有许许多多不同的神秘能量散布在这片大陆的各个角落，每份神秘能量都可以以一个正整数来描述其蕴含的能量值。年轻的暗影忍者认为这块神秘的大陆上的能量值为奇数的神秘能量数量若与能量值为偶数的神秘能量数量保持严格一致，那么这片大陆便是平衡的。

Input

第一行一个正正数 $T(T \leq 20)$ ，描述测试数据组数。

接下来每两行描述一组测试数据：

每组测试数据的第一行含有一个整数 $N(1 \leq N \leq 1024)$ 描述神秘大陆 HUXI 上的神秘能量数量；

每组测试数据的第二行含有 N 个正整数 $A_i(1 \leq A_i \leq 1024)$ 描述这些神秘能量所蕴含的能量值。

Output

对于每组测试数据，依次输出一个大写英文字母 'B' 表示神秘大陆 HUXI 是平衡的，或输出一个大写英文字母 'U' 表示大陆 HUXI 不平衡；

每组测试数据的输出独占一行(即每个大写英文之母之后需要换行，详如 Sample)。

Sample Input:	Sample Output:
2	B
4	U
1 2 3 1024	
4	
1 2 3 3	

Problem [B]rain Burst

time limit 2 seconds / memory limit 256 megabytes

Description

Brain Burst 2039 是日本轻小说作家川原砾创作的轻小说作品《加速世界》中的一款游戏的名称。

这是一款 VR-MMORPG(Virtual Reality - Massively Multiplayer Online Role-playing game)虚拟现实大型多人在线角色扮演类游戏，这款游戏与普通的 MMORPG 大型多人在线角色扮演类游戏有一个很重要的相似点：等级非常重要。

玩家刚进入游戏时等级为 Lv.1，当前经验值为 0 exp，当前等级升级需要经验值为 300 exp。玩家在游戏中获取经验将会累加在自己的当前经验值上，若一位 Lv 1 玩家获取经验后，一旦当前经验值不少于 300 exp 时便会升级，这时系统会立即在玩家的当前经验值上扣除玩家升级所需经验的，然后将玩家的等级提升至 Lv 2。

当然，升级所需经验会随着玩家等级的提升不断的升高，若当前玩家等级为 $K(K \geq 1)$ ，则玩家当前等级升级需要经验值为 $300 + (K - 1) * 100$ 。

然而在某些特定条件下，玩家甚至会连续升级：例如一名当前等级为 Lv 2，当前经验值为 200 exp 的玩家获取 1000 exp 经验后，那么他将会升级到 Lv 4，当前经验值剩余 300 exp (升级到 Lv 3 后当前经验剩余 800，不低于 Lv 3 到 Lv 4 的升级所需经验 500 故连续升级)。

由于游戏开发需要，现在给定某玩家获取 N exp 经验后的等级 Lv. L ，与该玩家获取经验后的当前经验值 E exp，需要编程计算求得这位玩家在获取这 N exp 经验之前的等级和经验值。

Input

第一行一个正整数 $T(T \leq 100)$ ，描述测试数据组数。

接下来每行描述一组测试数据：

每组测试数据唯一一行含有 3 个非负整数，依次为 $N(0 \leq N \leq 10000)$ 玩家获取经验， $L(1 \leq L \leq 99)$ 玩家获取经验后等级， $E(0 \leq E \leq 300 + (L - 1) * 100)$ 玩家获取经验后当前经验值。输入数据保证合法性（即玩家在获得经验前至少为 Lv 1, 0 exp）。

Output

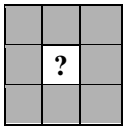
对于每组测试数据，依次输两个非负整数分别表示玩家在获取经验前的等级和经验值。

每组测试数据的输出独占一行(即每两个非负整数之后需要换行，详如 Sample)。

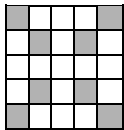
Sample Input:	Sample Output:
5	2 200
1000 4 300	1 0
300 2 0	1 100
100 1 200	4 300
0 4 300	2 315
1985 6 100	

Problem [C]ross Mines
time limit 2 seconds / memory limit 256 megabytes

Description
传统扫雷是一款单人的电脑游戏。游戏目标是找出所有没有地雷的方格，完成游戏；要是按了有地雷的方格，游戏失败。游戏以完成时间来评高低。
游戏开始时，玩家可看到一堆整齐排列的空白方块，方块数可由玩家自行选择。如果是第一次点开方块则不会踩到地雷。如果玩家点开没有地雷的方块，会有一个数字显现其上，这个数字代表着邻近方块有多少颗地雷（数字至多为 8），玩家须运用逻辑来推断哪些方块含或不含地雷。



传统扫示意图



交叉扫雷示意图

交叉扫雷与传统扫雷不同，没有地雷的方块上的统计数字统计的并非相邻方块上的地雷数量，而是统计以自己为正中心的 5*5 的正方形两条对角线上除自己之外的方格（之后略称为相对交叉位置）的地雷数量，如上图所示。
作为交叉扫雷的开发人员，你需要编程解决在输入所有地雷位置的情况下按照交叉扫雷规则标记所有没有地雷的方块其相对交叉位置方块中的地雷数量。

Input
第一行一个正正数 T(T<=20)，描述测试数据组数。
每组测试数据的第一行含有一个整数 N(1<=N<=16)描述交叉扫雷游戏棋盘尺寸为 N*N；
每组测试数据的第二行起总计 N 行每行共 N 个字符（小写字母‘x’表示该方块含有地雷，小写字母‘o’表示该方块没有地雷）。

Output
对于每组测试数据，依次输出一个 N*N 的字符矩阵对应每组输入数据。矩阵含有小写字母‘x’表示对应方格含有地雷或数字‘0’-‘8’表示对应方格其相对交叉位置方块中的地雷数量。

Sample Input:	Sample Output:
2	x1xx1
5	14xx2
xOXXO	x4324
oOXXO	x3xx1
XO00O	02x11
XOXXO	001
oOXOo	111
3	xx0
ooo	
ooo	
xxO	

Problem [D]Discount Shopping

time limit 2 seconds / memory limit 256 megabytes

Description

重庆大学 ACM-ICPC 代表队共 N 名队员共同前往 ACMOUNTAIN 出席比赛。赛前，大家偶然发现位于 ACMOUNTAIN 的 ACMARKET 正在进行大促销活动。

ACMARKET 的商品均分为可食用和不可食用两种，可食用商品全部 8 折优惠（原价 $\times 0.8$ ），不可食用商品原价。

若任意一个人购买了至少一件不可食用商品，那么他的所有所购买的商品中实际购买价格（即若是可食用商品将会以 8 折后价格计算）最低商品中的其中一件（即有多件商品实际购买价格最低时仅能优惠其中一件）会在实际购买价格的基础上再享受一次 5 折优惠（即若享受 8 折后的可食用商品是客户所购买的最低价的商品那么他可以选择以原价 $\times 0.8 \times 0.5$ 的价格来购买这件商品）。

代表队员们列出了他们希望购买的 M 件商品，请问 N 名队员最少共计使用多少钱可以购买该清单上的全部商品（每名队员最多仅能购买一次商品，但一次购买数量不受限制）。

Input

第一行一个正正数 T ($T \leq 20$)，描述测试数据组数。

接下来每行描述一组测试数据：

每组测试数据第 1 行两个整数，依次为 N ($1 \leq N \leq 10000$) 代表队队员个人, ($1 \leq M \leq 10000$) 清单商品个数；

每组测试数据第 2 至 $1+M$ 行每行两个整数，描述每一件在清单上的商品， P ($1 \leq p \leq 10000$) 商品原价， T 是否为可食用（ $T=0$ 时该商品为不可食用，反之 $T=1$ 时该商品为可食用）。

Output

对于每组测试数据，输出一个浮点数（保留两位小数，四舍五入）即购买清单上全部商品最少的花费。

每组测试数据的输出独占一行(即每个浮点数之后需要换行，详如 Sample)。

Sample Input:	Sample Output:
2	1587.20
2 3	2759.90
1024 0	
1024 1	
512 0	
2 5	
81 1	
511 0	
2137 0	
155 1	
1024 0	

Problem [E]agle Eye

time limit 2 seconds / memory limit 256 megabytes

Description

鹰眼是美国漫威漫画公司旗下的超级英雄。本名克林顿·法兰西斯·巴顿，小名克林特，曾化名为歌利亚与浪人，是个在马戏团长大的孤儿，师从剑客和捷射，天赋异常，少年时便获得“鹰眼”和“世界最佳狙击手”的称号，因在某次演出时看到钢铁侠救人的一幕，决心利用自己的能力成为超级英雄，后加入复仇者联盟，成为其中重要的一份子。在复仇者集结之前，鹰眼为神盾局工作。

在九头蛇的猛烈进攻下，鹰眼接到守卫一个神盾局秘密地下基地的任务：

秘密基地中有 N 个密室，这 N 个密室间共有 $N-1$ 条 1 个单位长度的通道相连，任意两个不同的密室都相互可达（或需要穿过其他的密室）。

神盾局需要鹰眼以及一些神盾局特工呆在部分密室中（不可移动）来保护这全部 N 个密室。

鹰眼由于其特殊的能力，他能够守卫自己所在的密室以及距离他所在密室不超过 K 个单位长度距离的所有密室。

其他特工能够守卫自己所在的密室以及距离他所在密室不超过 1 个单位长度距离的所有密室。

鹰眼希望你帮助他计算出要保护所有 N 个密室最少需要派遣多少名神盾局特工（鹰眼自身不计算在内）。

Input

第一行一个正整数 T ($T \leq 100$)，描述测试数据组数。

接下来每行描述一组测试数据：

每组测试数据第 1 行 2 个整数，依次为 N ($1 \leq N \leq 1000$) 表示密室数量， K ($1 \leq K \leq 5$) 表示鹰眼守卫范围；

每组测试数据第 2 至 N 行每行两个整数 X Y ，描述一条密室通道，说明密室 X 与密室 Y 间存在一条通道 ($1 \leq X, Y \leq N, X \neq Y$)，保证输入数据满足题意限制。

Output

对于每组测试数据，输出一个非负整数即最少派遣特工的数量。

每组测试数据的输出独占一行（即每个非负整数之后需要换行，详如 Sample）。

Sample Input:	Sample Output:
1 9 2 1 2 2 3 3 4 4 5 3 6 6 7 6 8 8 9	1

Problem [F]riends**time limit 2 seconds / memory limit 256 megabytes****Description**

These is an unusual social network. User can not add friends manually in this social network. Friends' relationship are determined by the system automatically in the following way: Assuming that user X had sent user Y a post at time T1, and user Y sent user X a message at time T2. If $T2 > T1$ and $T2 - T1 \leq D$, then system determine X and Y is a pair of friends because of that it seems the user Y replied the post which had been sent by X in time.

You are given the log of post-contains N post-records-in chronological order and a number D. Find the number of pairs of users who will be determined to be friends.

Input

The first line of the INPUT contain a positive integer T($T \leq 10$) which denotes the number of test cases.

The first line of each case contains two integers N and D ($1 \leq n, d \leq 1000$).

The next N lines of each case contain the post log. The i-th line contains one line of the log formatted as "Xi Yi Ti", which means that user Xi sent a post to user Yi at time Ti ($1 \leq i \leq n$). Xi and Yi are non-empty strings at most 20 characters long, consisting of lowercase letters, and Ti is an integer ($0 \leq Ti \leq 10000$). It is guaranteed that the lines are given in non-decreasing order of Ti's and that no user sent a message to himself. The elements in the lines are separated by single spaces.

Output

In each line print an integer k — the number of pairs of friends for each case.

Sample Input:	Sample Output:
3	1
4 3	0
jki curabis 7	2
curabis jki 8	
masoshonen jki 9	
jki masoshonen 15	
1 1000	
masoshonen masoshojo 0	
6 2	
a b 1	
b a 2	
a c 3	
a b 4	
c a 5	
c a 5	

Problem [G]raphic**time limit 2 seconds / memory limit 256 megabytes****Description**

Curabis always do excellent job on computation geometry problems. Indeed, he has acquired the first solution award about the computation geometry problems in the ACM-ICPC Asia Regional several times. However, JKi, his stupid teammate, do it much worse.

In order to help poor JKi, Curabis decide provide him with some practice problem set about computation geometry.

One of those problems is this:

Give a set of N 2d-points on the plane, is there any three 2d-points of the 2d-point set are able to constructive an triangle, and all 2d-points of this set are either inside of the triangle or on the edge of the triangle?

Would you please help JKi finishing this program?

Input

The first line of the INPUT contain a positive integer T ($T \leq 20$) which denotes the number of test cases.

The first line of each case contains an integer N .

The next N ($3 \leq N \leq 1000$) lines of each case contain the 2d-points set. The i -th line contains two real numbers X_i, Y_i denote the position of the 2d-point.

It is guaranteed that $-1000.00 \leq X_i, Y_i \leq 1000.00$, and all real numbers in the input keep two decimal places exactly.

It is guaranteed that all 2d-points in any case can not stand on one straight line overall.

Output

In each line print an upper-case letter — ‘Y’ for the triangle is existed, otherwise ‘N’ instead.

Sample Input:	Sample Output:
3	Y
4	N
0.00 0.00	Y
0.00 2.00	
2.00 0.00	
0.50 0.50	
4	
0.00 0.00	
0.00 2.00	
2.00 0.00	
2.00 2.00	
5	
0.00 0.00	
0.00 0.00	
1.00 0.00	
2.00 0.00	
1.00 1.00	