

NOIP 提高组复赛模拟（简单）

一、题目概况

中文题目名称	昆特牌	电梯	玩
英文题目与子目录名	gwent	lift	play
可执行文件名	gwent	lift	play
输入文件名	gwent.in	lift.in	play.in
输出文件名	gwent.out	lift.out	play.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	10	10
每个测试点分值	10	10	10
附加样例文件	无	无	无
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统
运行内存上限	256M	256M	256M

二、提交源程序文件名

对于 C++ 语言	gwent.cpp	lift.cpp	play.cpp
对于 C 语言	gwent.c	lift.c	play.c
对于 Pascal 语言	gwent.pas	lift.pas	play.pas

三、编译命令（不包含任何优化开关）

对于 C++ 语言	g++ -o gwent	g++ -o lift	g++ -o play
	gwent.cpp -lm	lift.cpp -lm	play.cpp -lm
对于 C 语言	gcc -o gwent	gcc -o lift lift.c	gcc -o play
	gwent.c -lm	-lm	play.c -lm

对于 Pascal 语言	fpc gwent.pas	fpc lift.pas	fpc play.pas
--------------	---------------	--------------	--------------

注意事项:

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
- 3、只提供 Linux 格式附加样例文件。
- 4、提交的程序代码文件的放置位置请参照各省的具体要求。
- 5、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

昆特牌（gwent）

题目背景

什么？你还有一场 NOIP 模拟赛？什么都别说，先来盘昆特牌吧！

题面描述

昆特牌是一款双人对战卡牌游戏，游戏规则现简化如下：

游戏简化为仅有一局；

每位牌手有三行，分别是近战、远程和攻城；

每位牌手有同样数量张牌，而聪明的你能预知对手每次出的牌！

牌分为以下几种功能：

单位牌：有一个战力数值和放置位置，**0**、**1**、**2** 分别表示近战、远程和攻城；

天气牌：有一个影响位置 **0**、**1**、**2** 分别表示近战、远程和攻城，对你的和对手的相应行都会影响。天气下的单位无法被烧灼，天气下的单位计算点数和的时候视为每个为 **1**，对已经有天气牌的行使用视作无效；

反天气牌：去除所有的天气牌并还原战力；

烧灼牌：消灭全场现有战力最高的单位（包括自己的单位）（除了天气下的），若有多个则都消灭，若没有单位则视为无效；

号角牌：选择 **6** 行（包括对手的）中一行使该行现有单位战力翻倍，但不能对天气牌影响下的行使用。如果每行都有天气视作无效，反之必须选择一行。由于丢了一些牌，这种卡最多只会出现 **2** 张，而且只有你有；

所有牌都必须使用，且除了单位牌其他牌都没有点数。视作无效的牌不影响整个出牌过程。你想要让你的点数和减去对手的尽量大（不是绝对值！），该怎么办呢？

输入数据

第一行两个数 **W**，**N** 分别表示先后手和每人牌数，**W=0** 表示你先手，**W=1** 表示对手先

接下来依次 N 行表示对手每次出的牌：

$t=1$ ，这是一张单位牌，接下来 x 、 y 分别表示战力和位置

$t=2$ ，这是一张天气牌，接下来 x 表示位置

$t=3$ ，这是一张反天气牌

$t=4$ ，这是一张烧灼牌

$t=5$ ，这是一张号角牌

接下来 N 行表示你的牌，描述同上。

输出数据

点数和差的最大值和点数和差最大方案数，用空格分隔。号角放在不同行算不同方案，且每张牌是不同的。例如你有两种方案：第一张号角放在你的 1 行第二张号角放在你的 0 行与第一张号角放在你的 0 行第二张号角放在你的 1 行不同。

样例数据 1

INPUT:

1 3

1 5 1

1 4 2

1 500 1

5

1 5 1

4

OUTPUT:

1 1

样例解释

唯一一种方案是：打出单位牌，打出号角选择自己的近战行，打出烧灼牌烧掉 500 点战力的牌，

这样你有 $5*2=10$ 点，对手有 $4+5=9$ 点

数据范围

对于 10% 的数据， $t=1$

对于 20% 的数据， $t=1$ 或 2

对于 40% 的数据， $t=1$ 或 2 或 3

对于 60% 的数据， $t=1$ 或 2 或 3 或 4

对于 100% 的数据， $N \leq 7$ ， $1 \leq \text{所有单位战力} \leq 1000$

电梯(lift)

题目背景

打完昆特牌后你不准备参加 NOIP 模拟赛了，因为你有更重要的赛事—NOGP（National Olympiad in Gwent in Provinces）你现在要上电梯去 N 楼参加 NOGP。在你从一楼上电梯的同时，你发现另有一台空电梯从一楼同时向 N 楼向上。

题面描述

N 层楼中有 M 层楼（除了 1 层、 N 层）被外面的人按了向上的按钮，电梯在遇到时将会停下 $1s$ 并且外面的人会在 $0s$ 内进入电梯。这样外面的按钮会被“吃掉”，也就是按钮消失且下一台电梯不会再停了。不幸的是由于你的电梯编号靠前，若两台电梯在同一有按钮的层时你的电梯会停下而另一台电梯会继续向上。在电梯里的你也可以按下一些内部的按钮使自己在某层停下 $1s$ （除了 1 层、 N 层）。你觉得输给一台空电梯非常丢脸，现在求在你不慢于空电梯的情况下最快在什么时候达到楼顶。若你不能不慢于空电梯，输出 -1 。电梯上一层楼所需时间为 $1s$ 。注意在一层楼你只能停一次。电梯不会超重什么的。

输入数据

第一行一个 T 表示数据组数

接下来每行五个数 N 、 M 、 a 、 b 、 c

$F(0) = c$

$F(i)$ = 第 i 个停靠楼层

$F(i) = (F(i-1) * a + b) \% (N-2) + 2$

如果 $F(i)$ 表示的楼层被按过了按钮，则 $F(i)$ = 从第二层往上第一个没有按过的按钮所在楼层

输出数据

对每行数据输出一行答案。

样例数据

INPUT

3

5 3 1 8 2

4 2 6 1 1

7 2 3 1 3

OUTPUT

6

4

7

数据范围

对于 20%的数据, $N \leq 10, M \leq 10$

对于 60%的数据, $N \leq 1000, M \leq 1000$

对于 100%的数据, $N \leq 10^7, M \leq 10^7, T \leq 10, N \geq M+2, a、b、c \leq 1000, N \geq 3$

玩 (play)

题目背景

你的昆特牌打的太好啦！不一会你就 AK 了 NOGP，只能无聊地堆牌玩！

题面描述

你有一些矩形卡牌，每次你会作如下三个操作：

- 1.往牌后面放一张牌，这张牌的底边与 X 轴重合且位于第一象限；（第一张牌最左边位于 $X=0$ 处）
- 2.拿走一张牌，并把后面的牌向前推；
- 3.询问 $[l, r]$ 这一段牌最高的高度；

输入数据

第一行两个数 $N、C$ 表示操作数、离线操作还是在线；

接下来 N 行每行是如下三种之一

$T=1$ ，接下来 $LEN、H$ 分别表示插入的牌的长、高；

$T=2$ ，接下来 X 表示拿走哪一张牌，牌的编号按出现顺序由小到大，从 **1** 开始（不是操作序号！），删除的牌不会改变序号，若序号表示的牌已被删除则不操作；

$T=3$ ，接下来 $L、R$ 表示询问区间；若询问区间上没牌，输出 **0**。

若 $C=1$ ，则上面输入中的 $LEN、H$ 需要以下式子算出（ $lastans$ 为最近一次询问答案）

真实输入 = (输入 * 2333 + $lastans * 666$) % 100000007 + 1;

输出数据

对每一个询问操作输出该段最高的位置高度是多少。

样例数据

INPUT

6 0
1 5 3
1 8 2
3 2 6
1 2 4
2 2
3 2 6

OUTPUT

3
4

样例解释

注意拿走一张牌时要把后面的牌向前推；

数据范围

对于 20% 的数据， $LEN=1$ ；

对于 40% 的数据，没有删除操作；

对于额外 20% 的数据， $C=0$ ；

对于 80% 的数据， $N \leq 2 \times 10^5$ ；

对于 100% 的数据， $N \leq 5 \times 10^5$ ， $LEN、H \leq 10^9$ 。

