

## 2015 年广东省重点中学信息学邀请赛（GDKOI 2015）

## 第二试

2015 年 3 月 1 日

注意事项：

1. 严格按照题目所要求的格式进行输入、输出，否则严重影响得分。
2. 题目测试数据有严格的时间限制，超时不得分。
3. 输入文件格式不用判错；输入输出文件名均已给定，不用键盘输入。
4. 源程序保存在以“序号+姓名”命名的文件夹下，按题目要求命名。评测以源程序为准。
5. 四个小时完成。
6. 内存限制：256MB，代码长度限制 32KB。
7. 本次竞赛的最终解释权归 GDKOI 评委会所有。

试题名称	看门狗	星球杯	V	复仇之魂
提交文件名	watchdogs.pas/c/cpp	planetcup.pas/c/cpp	v.pas/c/cpp	avenger.pas/c/cpp
输入文件名	watchdogs.in	planetcup.in	v.in	avenger.in
输出文件名	watchdogs.out	planetcup.out	v.out	avenger.out
满分	100	100	100	100

## 第一题 看门狗

提交文件：watchdogs.pas/c/cpp

输入文件：watchdogs.in

输出文件：watchdogs.out

## 题目描述：

艾登·皮尔斯最近接到了一些任务，该任务要求他在河的一岸设置一些干扰装置，干扰对岸一些设备的通讯。

皮尔斯需要干扰的设备有  $n$  个，排列在在了一边的河岸上，从南到北编号 1 到  $n$ ，皮尔斯在对岸租了  $n$  个房子，同样从南到北编号 1 到  $n$ ，由于地形问题，并不是每个房子都能干扰对面的所有设备。

当装置  $i$  被干扰，皮尔斯会得到  $P_i$  的报酬，若房间  $i$  设置了一个干扰装置，皮尔斯会得到  $V_i$  的经验值，每个干扰装置只能干扰一个设备。

由于干扰是通过无线信号实现的，所以同一个房间不能设置两个或以上的装置，并且干扰信号不能相交，例如：如果房间  $i, j$  都装有干扰装置，若  $i < j$  房间  $i$  的装置干扰设备  $S_i$ ，房间  $j$  的装置干扰  $S_j$ ，那么  $S_i$  必须小于  $S_j$ 。

皮尔斯对金钱以及经验都非常重视，所以希望这个任务得到的报酬和经验值的和最大。

## 输入格式：

输入样例第一行包含两个数： $n, m$ ， $n$  表示有  $n$  个房间以及  $n$  个需要被干扰的设备。

接下来有  $m$  行，描述可干扰关系。每一行两个数  $a, b$  表示在房间  $a$  设置干扰装置可以干扰到设备  $b$ 。

接下来一行  $n$  个数，第  $i$  个数表示干扰第  $i$  个设备能得到的报酬。

接下来一行  $n$  个数，第  $i$  个数表示在第  $i$  个房间设置装置能得到的经验值。

## 输出格式：

一个数表示皮尔斯能得到的最大的金钱经验和。

输入样例	输出样例
3 3 1 1 1 3 3 2 1 2 3 1 2 3	7

## 数据范围：

对于 30%数据： $1 \leq n \leq 100$

对于 100%数据： $1 \leq n \leq 100000$ ,  $0 \leq m \leq 500000$ ,  $0 \leq P_i, V_i \leq 1000$

# 第二题 星球杯

提交文件：planetcup.pas/c/cpp

输入文件：planetcup.in

输出文件：planetcup.out

## 问题描述

在北欧神话里，世界是由世界树所支撑,分成三层九个国度：

最上面一层：阿斯嘉德、华纳海姆、亚尔夫海姆；

中间一层：米德加尔特、约顿海姆、瓦特阿尔海姆；

最下面一层：海姆冥界、尼福尔海姆、穆斯贝尔海姆。

其中，阿斯嘉德和米德加尔特是相互密切往来的两个国度，两个国度通过彩虹桥连接起来。（看过电影的童鞋们，大概都知道阿斯嘉德国王往米德加尔特扔儿子和铁锤的故事吧）

除此之外，两个国度间也存在着不少民间团体的来往，其中最为著名的，就是一年一度的星球杯程序设计竞赛：阿斯嘉德和米德加尔特共派出  $N$  位选手参赛，比赛形式分为两轮预选赛和一轮决赛，每位选手必须且只能参加其中的一轮预选赛，并且对于每位选手，参加不同的预选赛会有不同的得分。最后，每轮预选赛中得分最高的前  $K$  名选手将能够进入最终的决赛（某轮比赛可以少于  $K$  人参加）。

赛事的安排由阿斯嘉德的二王子洛基负责，他可以指定这  $N$  位选手分别参加哪一场预选赛。在他看来，每位选手在预选赛里的得分表现，是能够影响选手在决赛里的最终得分的。所以，他希望所有晋级决赛的阿斯嘉德选手的得分总和最高（只取每位选手所参与预赛场次的得分）。

现在问题来了，你知道最高的得分总和是多少吗？

## 输入格式

第一行两个整数  $N$  ( $2 \leq N \leq 200$ ) 和  $K$  ( $1 \leq K \leq N/2$ )，分别表示参赛人数和每轮预选赛的晋级人数。

接下来  $N$  行，第  $i$  行为第  $i$  个选手的情况，每行三个整数，分别为该选手参加第一轮预选赛的得分  $X_i$ ，参加第二轮预选赛的得分  $Y_i$ ，以及该选手的所属国度  $Z_i$ （1 表示阿斯嘉德，0

表示米德加尔特)。

注意:  $X_i$  之间的值互不相同,  $Y_i$  之间的值互不相同。

## 输出格式

输出一行一个整数, 为阿斯嘉德最高的得分总和。

输入样例:	输出样例:
<pre> 5 2 90 60 0 40 50 1 3 1 1 4 4 0 1 2 1 </pre>	<pre> 54 </pre>

## 数据范围

对于 30% 数据,  $N \leq 20$ ;

对于 100% 数据,  $N \leq 200$ ,  $X_i$ ,  $Y_i$  的总和小于  $2^{31}$ 。

## 第三题 V

提交文件: v.pas/c/cpp

输入文件: v.in

输出文件: v.out

## 问题描述

给定一颗  $n$  个节点, 编号依次为 1 到  $n$  的树。要求依次回答  $q$  个询问, 每次询问  $u$  到  $v$  节点的最短路经过的点序列  $u \rightarrow \dots \rightarrow v$  中, 连续子序列  $a_1, a_2 \dots a_k$  满足  $a_1 < a_2 < a_3 \dots < a_j > a_{j+1} > a_{j+2} > \dots > a_k$  或者  $a_1 > a_2 > a_3 \dots > a_j < a_{j+1} < a_{j+2} < \dots < a_k$ ,  $1 \leq j \leq k$ , 求最大的  $k$ 。

## 输入格式

第一行有一个整数  $N$  表示点个数。

第二行  $N-1$  个数, 第  $i$  个数表示  $i+1$  号节点的父亲。1 号节点总是为根。

第三行一个整数  $Q$ , 表示询问个数。

接下来有  $Q$  行, 每行两个整数  $u, v$ 。(每次输入的  $u$  和  $v$  需要异或上次询问的结果, 得到真正的询问的  $u, v$  节点, 第一个询问异或 0)

## 输出格式

每个询问输出一行, 一个整数

输入样例:	输出样例:
<pre> Input1: 1  2 1 1 0 0  Input2: 2 1 2 1 1 0 3 </pre>	<pre> Output1: 1 1  Output2: 1 2 </pre>

## 数据范围

对于 20%数据,  $N, Q \leq 100$

对于 40%数据,  $N, Q \leq 20000$

对于 70%数据,  $N, Q \leq 50000$

对于 100%数据,  $N, Q \leq 100000$

## 第四题 复仇之魂

提交文件: `avenger.pas/c/cpp`

输入文件: `avenger.in`

输出文件: `avenger.out`

### 问题描述

小明最近十分喜欢玩塔防游戏。作为一名热爱游戏的程序员,他现在十分想自己做一款简单的塔防游戏《复仇者》。

游戏是这样的,玩家控制着一座城池,可以合理地建设防御措施,来抵御怪物的入侵,在玩家建立好防御措施之后,怪物入侵就开始了。值得一提的是,这个游戏不同于以往的游戏,在怪物入侵之后就不能再建设防御措施。

怪物入侵有无数轮(无尽模式)。简单起见,小明只设计了一种怪物“复仇之魂”。因为小明是生活在一个  $K$  进制的世界下,所以每一轮怪物入侵,他都设定有  $K$  只“复仇之魂”。

怪物血量的设定方法是这样的:有一个无穷大的栈(在游戏开始时先在栈里放置了无数个  $0$ )。每一轮怪物入侵,就取出栈顶的数,分成  $K$  等份,作为这一轮各个“复仇之魂”的血量。

如果玩家成功把所有怪物杀死,就成功守下了这一波怪物入侵;一旦玩家没能够把一轮的怪物杀光,就会有怪物进入到玩家的城池中,玩家就会少一条命。

“复仇之魂”是一种特殊的怪物,一旦将一只怪物杀死,就会积累等值于他的血量的仇恨值,另外每只怪物的死亡会产生额外的  $A$  点仇恨值。如果成功守下一轮,将产生新的  $K$  轮怪物,并将仇恨值复制  $K$  份置于栈顶。每轮怪物入侵之后,不管成功与否,仇恨值都会清空。

由于小明在写代码的时候不小心把栈实现错了,导致每一次从栈顶取出一个数之后,栈里的所有数,在  $k$  进制的表示下的最低位都会出现循环变换(假设最低位原来为  $X$ ,那么现在变为  $(X+1) \bmod K$ ,即  $0$  变成  $1, 1$  变成  $2, \dots, k-1$  变成  $0$ )。这样导致了有可能栈顶数字不能平分成  $k$  等份,那么怪物的血量设定也会发生改变。假设现在栈顶的值为  $X$ ,此时就会出

现  $k-1$  个怪物的血量为  $\lfloor X/k \rfloor$ ,还有  $1$  个怪物的血量为  $X - (k-1)\lfloor X/k \rfloor$  ( $\lfloor x \rfloor$  表示  $x$  向下取整)。

在完成这个游戏之后,小明决定自己先玩玩看。小明建设的城池可以给怪物总共造成  $L$  点伤害(可以任意分配伤害到各个怪物身上),并且小明游戏开始的时候有  $X$  条命。现在小明想知道把自己击败的最后一轮的怪物的总血量是多少。

### 输入格式

第一行输入  $T$ ,表示有多少个测试数据。

接下来  $T$  行，每一行代表一个测试样例。

每一行的格式如下：

$X, K, L, A$

$X$  表示玩家有多少条命， $K$  表示进制， $L$  表示玩家每一轮怪物入侵能够对怪物造成的伤害， $A$  表示每只怪物额外增长的仇恨值。

## 输出格式

输入样例：	输出样例：
4 1 2 10 6 2 2 10 6 3 2 10 6 8 2 7 3	12 13 13 13

## 数据范围

对于 10% 数据， $1 \leq T \leq 100$ ， $1 \leq X, L, A \leq 1000$ ， $K=2$

对于 30% 数据， $1 \leq T \leq 100$ ， $1 \leq X, L, A \leq 1000$ ， $K=2$

对于 50% 数据， $1 \leq T \leq 100$ ， $1 \leq X \leq 100,000,000$ ， $1 \leq L, A \leq 100,000$ ， $2 \leq K \leq 100,000$

对于 100% 数据， $1 \leq T \leq 300000$ ， $1 \leq X, L, A \leq 100,000,000$ ， $2 \leq K \leq 100,000,000$

## 提示

前三组样例的解释：（下图为每一轮结束时栈的情况）

第一轮：游戏开始时，第一波怪物为 2 个 0 血的“复仇之魂”，出场之后直接死了。此时总共累积了 12 点仇恨值，将 2 个 12 放进栈顶之后。

第二轮：怪物为 2 个 6 血的“复仇之魂”，由于玩家造成的伤害杀不死所有怪物 ( $2 \times 6 > 10$ )，玩家掉一条命。此时产生了仇恨值，但是由于没有能够把所有怪物杀死，并不会将仇恨值加入栈内。

第三轮：怪物为 1 个 6 血、1 个 7 血的“复仇之魂”，同样无法成功守住 ( $6+7 > 10$ )，再掉一条命。

第四轮：怪物为 1 个 0 血、1 个 1 血的“复仇之魂”，成功守住。积累了 13 点仇恨值，再将 2 个 13 放进栈顶。

第五轮：怪物为 1 个 6 血，1 个 7 血的“复仇之魂”，无法守住，掉一条命。

