



# 第二十届“程协杯”程序设计竞赛

2025 年 11 月 16 日，成都理工大学

Produced by 第二十届“程协杯”命题组

2025 年 11 月 16 日

## 目录

<b>Problem A. 对称</b>	<b>2</b>
<b>Problem B. 拼手速</b>	<b>3</b>
<b>Problem C. 非成都电路板</b>	<b>4</b>
<b>Problem D. 线段树？</b>	<b>5</b>
<b>Problem E. 古希腊掌管三角形的神</b>	<b>6</b>
<b>Problem F. 水 + 水 = 水</b>	<b>7</b>
<b>Problem G. 恋爱物语</b>	<b>9</b>
<b>Problem H. 拼图</b>	<b>10</b>
<b>Problem I. 像素水果忍者</b>	<b>11</b>
<b>Problem J. 我不是药神</b>	<b>12</b>
<b>Problem K. 上进的冰妖精</b>	<b>13</b>
<b>Problem L. 《世界》</b>	<b>14</b>
<b>Problem M. 晚餐</b>	<b>17</b>

## Problem A. 对称

埃及法老 Aiji Falao 是一位暴君，他喜欢抓人来解决自己出的谜题，成功解决能获得一定的赏金，但如果无法解决就会被当作祭品。很不幸，这天，你被抓住了；但是很幸运，你的大脑连接到了第 114154 届程协杯比赛现场的某台电脑上。

Aiji Falao 向你发起了询问：

我有一个  $n \times n$  的棋盘和  $n$  个相同的棋子。每一行、每一列最多放一个棋子，而且棋子摆放的形状必须关于主对角线（从左上到右下）对称。告诉我，棋子有多少种摆放的方式？

为了避免成为祭品，你需要立即编程计算出有多少种摆法。

### 输入描述

第一行一个整数  $T(1 \leq T \leq 10^6)$  表示数据组数。

接下来  $T$  行，每行一个整数  $n(1 \leq n \leq 10^6)$  表示棋盘大小和土块个数。

### 输出描述

对于每组数据，输出一行一个整数，表示摆法的数量。

由于答案可能非常大，所以答案对  $10^9 + 7$  取模。

### 样例

Input 1	Output 1
3	1
1	2
2	4
3	

### 说明/提示

如果存在位置  $(i, j)$ ，使得两种摆法在  $(i, j)$  上一个有棋子而另一个没有，那么这两种摆法是不同的。

当  $n = 1$  时，摆法有  $(1, 1)$ 。

当  $n = 2$  时，摆法有  $(1, 1)(2, 2)$ 、 $(1, 2)(2, 1)$ 。

当  $n = 3$  时，摆法有  $(1, 1)(2, 2)(3, 3)$ 、 $(1, 1)(3, 2)(2, 3)$ 、 $(1, 3)(2, 2)(3, 1)$ 、 $(1, 2)(2, 1)(3, 3)$ 。

## Problem B. 拼手速

6111 年 52 月 02 日，小 M 参加了第 114514 届程协杯。这次程协杯持续时间为  $10^{10^{10}}$  分钟（以下默认时间单位为分钟），有  $n$  道题目，采用 CIALLO 赛制。

CIALLO 赛制和大家熟知的 ACM 赛制相似，不同的是，CIALLO 赛制没有罚时！每道题目的用时为比赛开始到该题目被正确解决的时间，最终，选手的总用时是所有被正确解决的题目的用时的和。

每天刷水题的小 M 瞬间看完了所有的题目，并且算出了每道题目要花费的时间。他立刻开始解题，每道题目解出后立刻开始解下一题，保证不会出错。

小 M 可以以任意顺序解题，请求出小 M 的总用时最低是多少。

### 输入描述

第一行一个整数  $T(1 \leq T \leq 100)$  表示数据组数，接下来按组给出数据。

对于每组数据：

第一行一个整数  $n(1 \leq n \leq 20)$  代表程协杯的题目数量；

随后一行  $n$  个整数  $t_1, t_2, \dots, t_n(0 \leq t_i \leq 1000)$ ， $t_i$  代表第  $i$  道题目花费的时间。

### 输出描述

对于每组数据，输出一行一个整数，代表小 M 最低的总用时。

### 样例

#### Input 1

```
2
2
2 1
5
1 1 1 1 1
```

#### Output 1

```
4
15
```

### 说明/提示

对于第一组数据：最优的解题顺序是 [2, 1]，用时为 [1, 3]，总用时为 4。

对于第二组数据：解题顺序是 [1, 2, 3, 4, 5]，用时为 [1, 2, 3, 4, 5]，总用时为 15。

显然，选手应当先解决花费时间少的题目。

## Problem C. 非成都电路板

6Jun 来到成都，打算在这座慢节奏又卷得离谱的城市里搞点新东西。他正调试一块电路板。电路板可以视为由芯片组成的  $n \times m$  的矩阵  $A$ ，第  $i$  行第  $j$  个芯片的参数为  $A_{i,j}$ 。但是问题来了：

6Jun 讨厌一切「同」。

同样的想法、同样的咖啡、同样的 Bug、还有最要命的——相邻芯片的值也「同」！一旦两块相邻的芯片参数一样，整个系统就陷入“同化危机”：电路发烫、风扇狂转。这个危机会让 6Jun 直接红温！

为了避免出现“同化危机”，6Jun 决定动手修改芯片参数。他可以进行一次如下操作：选择一些芯片，让这些芯片的参数  $+x$ 。但是，操作只能进行一次，于是他找到了聪明的你并且希望你能够帮他解决这个难题。形式化地，你需要设计一个新的矩阵  $B$ ，让它符合以下要求：

对于所有的位置  $(i, j)$ ， $B_{i,j} = A_{i,j}$  或  $B_{i,j} = A_{i,j} + x$ ，其中  $1 \leq i \leq n, 1 \leq j \leq m$ 。

不存在位置  $(i, j)$  使得  $B_{i,j} = B_{i+1,j}$  或  $B_{i,j} = B_{i,j+1}$ ，其中  $1 \leq i < n, 1 \leq j < m$ 。<sup>1</sup>

### 输入描述

第一行包一个整数  $T$  ( $1 \leq T \leq 100$ ) 代表数据组数，接下来按组给出数据。

对于每组数据：

第一行三个整数  $n, m, x$  ( $1 \leq n, m \leq 100, 1 \leq x \leq 10^4$ ) 分别表示行数、列数、修改值。

接下来  $n$  行，每行  $m$  个整数，第  $i$  行第  $j$  个表示  $A_{i,j}$  ( $0 \leq A_{i,j} \leq 10^8$ )。

### 输出描述

对于每组数据，输出  $n$  行，每行  $m$  个整数。第  $i$  行第  $j$  个表示  $B_{i,j}$ 。

### 样例

Input 1	Output 1
2	4 3 2 1 2 3
1 6 1	2 3
3 2 1 1 2 3	3 4
3 2 1	4 5
2 2	
3 3	
4 4	

### 说明/提示

对于第一组数据：选择  $(1, 1), (1, 2), (1, 3)$ 。对于第二组数据：选择  $(1, 2), (2, 2), (3, 2)$ 。

<sup>1</sup>当所有芯片都各有个性、再也没有「同」的那一刻，6Jun 坐在成都的夜色里，喝着蜜雪冰城，露出满意的笑：“哈哈哈，终于没有‘同’了。”

## Problem D. 线段树？

小曾和小张是两名数学爱好者。小曾有一个正整数  $X$ ，小张有一个长度为  $n$  的序列  $a$ 。现在，他们决定测试一下他们之间的默契度。小曾提出了这样一种方式：

每次从小张的序列中随机选择一个区间  $l, r$  并计算区间中所有元素的乘积  $val_{l,r}$ ，然后找到最大非负整数  $k$ ，使得小曾手中的整数  $X$  的  $k$  次方能够整除  $val_{l,r}$ 。将  $k$  记为本次的得分。

现在，他们找到了作为编程高手的你来帮他们计算得分并输出。形式化地，对于每次询问的  $l, r$ ，求出最大的非负整数  $k$  使得：

$$\left( \prod_{i=l}^r a_i \right) \bmod X^k = 0$$

### 输入描述

第一行两个整数  $n, X(1 \leq n \leq 10^5, 2 \leq X \leq 10^9)$  表示小张的序列的元素的个数和小曾的正整数。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq 10^9)$  表示小张的序列  $a$ 。

第三行一个整数  $m(1 \leq m \leq 10^5)$  表示询问的次数。

接下来  $m$  行每行两个整数  $l, r(1 \leq l \leq r \leq n)$  表示询问的区间。

### 输出描述

对于每个询问，输出一行一个整数表示这个询问的得分  $k$ 。

### 样例

Input 1	Output 1
3 114514	2
114514 10 114514	0
2	
1 3	
2 2	

### 说明/提示

对于区间  $[1, 3]$ ， $val_{l,r} = 114514^2 \times 10$ ，可以被  $114514^2$  整除但是不能被  $114514^3$  整除，故  $k$  为 2。

对于区间  $[2, 2]$ ， $val_{l,r} = 10$ ，可以被  $114514^0$  整除但是不能被  $114514^1$  整除，故  $k$  为 0。

## Problem E. 古希腊掌管三角形的神

秧歌(又被称为年轻的云)是古希腊掌管三角形的神,无所畏惧的你来到秧歌的面前想要挑战他的权威,秧歌不屑一顾地对你说:“何意味?正确回答这个问题你才有资格触碰神圣的三角形”。

秧歌将  $n$  根木棍摆在你的面前,第  $i$  根木棍的长度为  $a_i$ 。接着他会提出  $q$  个对木棍的查询,每个查询会给出两个整数  $l, r$  ( $1 \leq l < r \leq n$ , 且  $r - l + 1 \geq 6$ )。你需要回答,能否从第  $l$  到  $r$  (包含  $l, r$ ) 个木棍中,挑选出 6 根不同的木棍,组成 2 个非退化三角形(若一个三角形的三边长  $a b c$  满足  $a < b + c$  且  $b < a + c$  且  $c < a + b$ ,则称其为非退化三角形)。

### 输入描述

第一行两个整数  $n$  和  $q$  ( $6 \leq n \leq 10^5$   $1 \leq q \leq 10^5$ ) 分别表示木棍的数量和查询的数量。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) 表示木棍长度。

接下来  $q$  行,每行两个整数  $l, r$  ( $1 \leq l < r \leq n$ , 且  $r - l + 1 \geq 6$ ) 表示询问。

### 输出描述

对于每个查询,若能组成 2 个非退化三角形,输出一行 YES,否则输出一行 NO。

### 样例

Input 1	Output 1
10 5	YES
5 2 2 10 4 10 6 1 5 3	NO
1 6	YES
2 7	NO
2 8	YES
5 10	
4 10	

### 说明/提示

第一个查询中,木棍长度为  $[5, 2, 2, 10, 4, 10]$ ,可选择两组木棍  $[2, 4, 5]$  和  $[2, 10, 10]$ ,分别组成 2 个非退化三角形。

第二个查询中,木棍长度为  $[2, 2, 10, 4, 10, 6]$ ,无法组成 2 个非退化三角形。

第三个查询中,木棍长度为  $[2, 2, 10, 4, 10, 6, 1]$ ,可选择两组木棍  $[1, 2, 2]$  和  $[4, 10, 10]$ ,分别组成 2 个非退化三角形。

第四个查询中,木棍长度为  $[4, 10, 6, 1, 5, 3]$ ,无法组成 2 个非退化三角形。

第五个查询中,木棍长度为  $[10, 4, 10, 6, 1, 5, 3]$ ,可选择两组木棍  $[1, 10, 10]$  和  $[3, 4, 5]$ ,分别组成 2 个非退化三角形。

## Problem F. 水 + 水 = 水

在一个二维网格世界中，水超现实地流动着。网格大小为  $m \times n$ ，左上角为  $(0, 0)$ ，右下角为  $(m, n)$ 。

「水源」：水流的中心。

「流动」：水会以水源为中心向周围流动，每流动一格，流量-1。流量降至 0 时，水不再继续流动。

「流量」：一个格子可能会被多处水源流动到，该格子的流量为所有水源流动到此处的流量的最大值。水源的流量为 5。

「水的生成」：当一个格子周围四个存在至少两个水源，则该格子变为水源。水源会不断生成，直到没有格子满足生成水源的条件。

例如，水流的最终状态可能是这样：

0	0	0	0	1	0	0	1	0	0	0	0	0
0	0	0	1	2	1	1	2	1	0	0	0	0
0	0	1	2	3	2	2	3	2	1	0	0	0
0	1	2	3	4	3	3	4	3	2	1	0	0
1	2	3	4	5	4	4	5	4	3	2	1	0
0	1	2	3	4	3	3	4	3	2	1	0	0
0	0	1	2	3	2	2	3	2	1	0	0	0
0	0	0	1	2	1	1	2	1	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	0	0

图 1: 水的流动

如图2所示，被标记的格子周围存在两个水源，因此会形成新的水源。

0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	1	2	1	2	1	0	0	0	0	0
0	0	1	2	3	2	3	2	1	0	0	0	0
0	1	2	3	4	3	4	3	2	1	0	0	0
1	2	3	4	5	4	5	4	3	2	1	0	0
0	1	2	3	4	3	4	3	2	1	0	0	0
0	0	1	2	3	2	3	2	1	0	0	0	0
0	0	0	1	2	1	1	2	1	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	0	0

图 2: 中间状态

新的水源生成后，最终状态如图3所示。

0	0	0	0	1	1	1	0	0	0	0
0	0	0	1	2	2	2	1	0	0	0
0	0	1	2	3	3	3	2	1	0	0
0	1	2	3	4	4	4	3	2	1	0
1	2	3	4	5	5	5	4	3	2	1
0	1	2	3	4	4	4	3	2	1	0
0	0	1	2	3	3	3	2	1	0	0
0	0	0	1	2	2	2	1	0	0	0
0	0	0	0	1	1	1	0	0	0	0

图 3: 新的水源生成

世界创立之初， $k$  个格子上存在水源，随之，水开始流动、生成。请你计算最终每个格子上的流量。

### 输入描述

第一行一个整数  $T(1 \leq T \leq 100)$  表示数据组数，接下来按组给出数据。

对于每组数据：

第一行包含三个整数  $m, n, k$  ( $1 \leq m, n \leq 20$ )，表示网格的行数和列数、初始水源数量。

接下来  $k$  行，每行包含两个整数  $x_i, y_i$  ( $0 \leq x_i < m, 0 \leq y_i < n$ ) 表示初始水源的坐标。保证坐标不会重复。

### 输出描述

对于每组数据，输出  $m$  行，每行包含  $n$  个整数，表示最终每个格子的流量。

### 样例

Input 1	Output 1
1	554321
4 6 2	554321
0 0	443210
1 1	332100

### 说明/提示

在  $(0, 1)$  和  $(1, 0)$  处生成了新的水源。

## Problem G. 恋爱物语

广袤的互联网中有这样一个 QQ 群，其中不少人向往甜甜的恋爱。

令：想要甜甜的恋爱怎么办（2025-11-04 09:49）

涵：帮我算算我什么时候可以谈恋爱（2025-11-04 20:52）

werwer：想谈甜甜的恋爱了（2025-11-03 18:23）

明末美食家：我也想了解一下恋爱的细节（2025-10-29 23:03）

QQ 群中有  $n$  个男生和  $m$  个女生，每个人有且仅有一个喜欢的异性。如果两个人互相喜欢，他们会成为一对情侣。

群主小 M 想知道 QQ 群中现在有多少对互相喜欢的关系。

### 输入描述

第一行一个整数  $T(1 \leq T \leq 100)$  表示数据组数，接下来按组给出数据。

对于每组数据：

第一行两个整数  $n, m(2 \leq n, m \leq 100)$  代表男生、女生人数。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq m)$ ，第  $i$  个男生喜欢第  $a_i$  个女生。

第三行  $m$  个整数  $b_1, b_2, \dots, b_m(1 \leq b_i \leq n)$ ，第  $i$  个女生喜欢第  $b_i$  个男生。

### 输出描述

对于每组数据，输出一行一个整数，表示相互喜欢关系的数量。

### 样例

#### Input 1

```
2
4 5
1 1 3 3
1 2 4 3 1
5 4
1 2 3 4 4
3 2 2 1
```

#### Output 1

```
2
1
```

### 说明/提示

第一组数据，互相喜欢的关系有  $(1, 1), (4, 3)$ 。

第二组数据，互相喜欢的关系有  $(2, 2)$ 。

## Problem H. 拼图

小 C 有一个特别的拼图，拼图有以下四种组件：

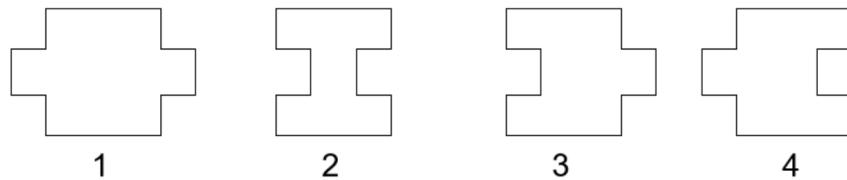


图 4: 四种组件

每个组件的左右两侧各有一个连接结构：要么是凸起，要么是凹陷。组件不可旋转。

两个组件能够连接的条件是：左侧组件的右侧连接结构与右侧组件的左侧连接结构互为相反类型（即凸起与凹陷配对）。

拼图中每种类型的组件数量分别为  $c_1, c_2, c_3, c_4$ （数量按照图片中的顺序给出）。若能将所有组件拼接成一条完整的长链，则认为拼图完成。

你需要帮助小 C 求出有多少种不同的拼接方式。

### 输入描述

第一行一个整数  $T(1 \leq T \leq 2 \times 10^5)$  表示数据的组数。

接下来  $T$  行，每行四个整数  $c_1, c_2, c_3, c_4(0 \leq c_i \leq 10^6)$  表示每种组件的数量。

保证所有测试用例的  $c_i$  之和不超过  $4 \times 10^6$ 。

### 输出描述

对于每组数据，输出一行一个整数，表示完成拼图的方案数。

由于答案可能非常大，所以答案对 998244353 取模。

如果无法完成这个拼图，输出 0。如果  $c_1 = c_2 = c_3 = c_4 = 0$ ，方案数为 1。

### 样例

Input 1	Output 1
5	1
0 0 0 4	4
1 1 1 1	36
5 5 0 2	0
4 6 8 10	794100779
900000 900000 900000 900000	

### 说明/提示

如果存在  $i$  使得两种方案的第  $i$  个组件不同，那么这两种方法是不同的。

## Problem I. 像素水果忍者

年轻的云（又被称为秧歌）最近迷上了一款叫作《水果忍者》的游戏。在这个游戏中，屏幕上会出现水果，你需要在屏幕上滑动，如果滑动的路径和水果有重合的部分，你就能切开这个水果并获得分数。

由于这个游戏只能免费玩 114514 秒，秧歌决定自己开发一款新游戏《像素水果忍者》。在这里，水果都是矩形的且不会倾斜，而你画出路径都是线段。秧歌想让你帮他完成判断水果是否被切开的功能。

游戏可以抽象为一个平面直角坐标系。每次，给出一条线段的左端点  $(x_1, y_1)$  和右端点  $(x_2, y_2)$ ，如果线段垂直于  $x$  轴则按任意顺序给出；给出一个矩形的左下角  $(x_l, y_l)$  和右上角  $(x_r, y_r)$ ，保证四边平行于坐标轴；判断线段和矩形有没有交集。

### 输入描述

第一行一个整数  $T(1 \leq T \leq 10^5)$  表示数据组数。接下来按组给出数据。

对于每组数据：

第一行四个整数  $x_1, y_1, x_2, y_2$  表示线段的左右端点。

第二行四个整数  $x_l, y_l, x_r, y_r$  表示矩形左下角和右上角。

保证给出的坐标值的绝对值不超过  $10^4$ 。保证线段不会退化成点，矩形不会退化成线段或点。

### 输出描述

对于每组数据，如果有交集则输出一行 YES，否则输出一行 NO。

### 样例

Input 1	Output 1
3	YES
0 0 5 5	NO
0 0 10 10	YES
0 0 5 5	
6 6 7 7	
0 0 5 5	
1 1 2 2	

### 说明/提示

如果存在一个点既在线段上（含端点）也在矩形中（含边界），则认为线段和矩形有交集。

## Problem J. 我不是药神

Alice 和 Bob 开了一家药店。这天他们进货了  $n$  瓶药，每瓶药中有  $n$  粒药丸。Alice 在检查这批药的时候，发现这  $n$  瓶药中有且仅有 1 瓶是假药，同时她发现，如果一瓶药是真药，则里面的每粒药丸都重 500mg；如果一瓶药是假药，则里面的每粒药丸都重 510mg。Alice 告诉 Bob 有一瓶是假药，也告诉了 Bob 真药和假药的区别。她想考考 Bob，于是规定了一种称重方式：

从每瓶药中取出任意粒药丸，一并称出取出的药丸的总重量。

形式化地，你可以任意构造一个数组  $a_1, a_2, \dots, a_n (0 \leq a_i \leq n)$ ，从第  $i$  瓶中取出  $a_i$  粒药丸并消耗一次称重次数得到这  $\sum_{i=1}^k a_i$  粒药丸的总重量。

Alice 问 Bob，在她给定的称重方式下，至少要称重多少次才能确定第几瓶药是假的。但是 Bob 一时想不出来，于是找了你来帮忙。

### 输入描述

第一行一个整数  $T (1 \leq T \leq 10^5)$  表示数据组数。

随后  $T$  行，每行一个整数  $n (1 \leq n \leq 10^9)$ ，表示药的瓶数。

### 输出描述

对于每组数据，输出一行一个整数表示最少称重次数。

### 样例

Input 1	Output 1
2	0
1	1
2	

### 说明/提示

如果有两瓶药，我们只从某一瓶中取一粒药丸并称重。若该药丸重 510mg，则对应的那瓶药为假；否则，另一瓶为假。称重一次即可，答案为 1。

## Problem K. 上进的冰妖精

小⑨是一只冰妖精。她十分上进，讨厌一切下降的东西。这天，她来到雾之湖边玩耍，发现了一排冰柱，第  $i$  根冰柱的高度为  $h_i$ 。但是，这些冰柱参差不齐，也就是，其中可能含有「下降」！

小⑨认为，如果存在  $i(1 \leq i < n)$  使得  $h_i > h_{i+1}$ ，那么这些冰柱是不上进的。否则，这些冰柱是上进的。作为冰妖精，小⑨有操纵冰程度的能力。每次，她可以选择一些冰柱，并且让它们的高度-1。

最少需要使用多少次能力才能让这些冰柱变得上进呢？作为大数学家的小⑨并不屑于计算这么简单的问题，于是把这个任务交给了你。

### 输入描述

第一行一个整数  $n(1 \leq n \leq 10^6)$  表示序列元素的个数。

第二行  $n(1 \leq h_i \leq 10^9)$  个整数，表示冰柱的高度。

### 输出描述

输出一行一个整数，表示最少的使用能力的次数。

### 样例

Input 1	Output 1
3	1
1 3 2	

### 说明/提示

选择第二根柱子并使其高度-1，冰柱的高度变为 [1,2,2]，满足条件。故使用一次能力即可，答案为 1。

## Problem L. 《世界》

小 M 喜欢玩一款叫作《世界》的游戏。

### 构成与法则

《世界》是一个由方块和地面组成的三维空间，其中，地面是一种由特殊的同种方块构成的无限大平面，甚至可以延伸到《世界》外部；方块一共有三种，分别是沙子、石头和火把。我们可以将《世界》的大小表示为  $X, Y, Z$ ，将内部的位置集合表示为

$$\{(x, y, z) | x \in [0, X - 1], y \in [0, Y - 1], z \in [0, Z - 1]\}$$

将地面表示为

$$\{(x, y, z) | z = -1\}$$

《世界》中有一套特殊的物理法则：

方块可以被放置某位置，当且仅当该位置在《世界》内部，且该位置当前没有方块，且相邻六个面的位置已存在方块或为地面。

当方块的下方没有紧贴着地面或者方块，我们认为这个方块是悬浮的。石头可以悬浮而不会落下；沙子悬浮会下落。

火把是一种特殊的方块，只能附着在相邻的非火把的方块（包括地面）的侧面或上面，且附着关系一旦建立就不会改变。当火把附着的方块被破坏/下落，该火把立刻消失；其余情况下火把不会改变状态。

显然，当没有沙子悬浮的时候，《世界》是稳定的。

### 你的任务

小 M 在心中构想了一份计划。计划分为  $N$  条指令，每一条可能是放置或者破坏。他想让你检测计划的可行性，因此，你需要按顺序检测每条指令是否合法。

如果当前指令合法，执行之，输出一行 GOODJOB，等待《世界》稳定后再进行下一步；否则，跳过当前指令，输出一行 AREYOUKIDDINGME。所有指令检测完毕后，输出《世界》的状态，具体输出方法将在下文说明。

### 指令种类

**放石头/沙子：**输入格式为一个字符串和三个整数： $PUT\_STONE\ x\ y\ z$  或  $PUT\_SAND\ x\ y\ z$ ，参数表示放置位置。如果放置位置没有方块且相邻六个面的位置存在方块或为地面，则合法。

**插火把：**输入格式为一个字符串和四个整数， $PUT\_TORCH\ x\ y\ z\ f$ ，参数表示放置位置和附着的方向。如果放置位置没有方块且附着位置上存在不为火把的方块或为地面，则合法。 $f$  有以下几种取值：

- 1: 附着在坐标为  $(x + 1, y, z)$  上；
- 2: 附着在坐标为  $(x - 1, y, z)$  上；
- 3: 附着在坐标为  $(x, y + 1, z)$  上；

- 4: 附着在坐标为  $(x, y - 1, z)$  上；  
 5: 附着在坐标为  $(x, y, z - 1)$  上。

**破坏：**输入格式为一个字符串和三个整数， $DESTROY\ x\ y\ z$  参数表示破坏位置。如果破坏位置上存在方块，则合法。

## 状态输出

你需要按照下方伪代码表示的方式输出《世界》状态：

```

OUTPUT "THE WORLD"
NEWLINE
FOR k FROM 0 TO Z-1
  FOR i FROM 0 TO X-1
    FOR j FROM 0 TO Y-1
      IF stone on (i,j,k) THEN
        OUTPUT 'r'
      ELSE IF sand on (i,j,k) THEN
        OUTPUT 's'
      ELSE IF torch on (i,j,k) THEN
        IF f IS 1 THEN
          OUTPUT 'v' (小写字母 v)
        ELSE IF f IS 2 THEN
          OUTPUT '^' (shift+6)
        ELSE IF f IS 3 THEN
          OUTPUT '>' (shift+ 句号)
        ELSE IF f IS 4 THEN
          OUTPUT '<' (shift+ 逗号)
        ELSE
          OUTPUT 't'
        ENDIF
      ELSE
        OUTPUT 'a'
      ENDIF
    ENDFOR
    NEWLINE
  ENDFOR
  NEWLINE
ENDFOR

```

## 输入描述

第一行三个整数  $X, Y, Z (1 \leq X, Y, Z \leq 10)$  表示《世界》的大小。

第二行一个整数  $N (1 \leq N \leq 10^5)$  代表指令数量。

随后  $N$  行，每行一条指令，具体格式见题目描述。

对于所有指令，保证  $0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z$ 。

## 输出描述

对于每条指令，输出一行 GOODJOB 或仅 AREYOUKIDDINGME。  
最后输出《世界》的状态。

## 样例

<b>Input 1</b>	<b>Output 1</b>
2 2 2	AREYOUKIDDINGME
6	GOODJOB
PUT_SAND 0 0 1	GOODJOB
PUT_STONE 1 0 0	GOODJOB
PUT_TORCH 0 0 0 1	AREYOUKIDDINGME
PUT_SAND 0 0 1	GOODJOB
DESTROY 0 1 0	sa
DESTROY 1 0 0	aa
	aa
	aa
	aa

## 说明/提示

样例演示了：放置石头，在石头侧面附着火把，在火把上放沙子；破坏石头，火把消失，沙子下落。

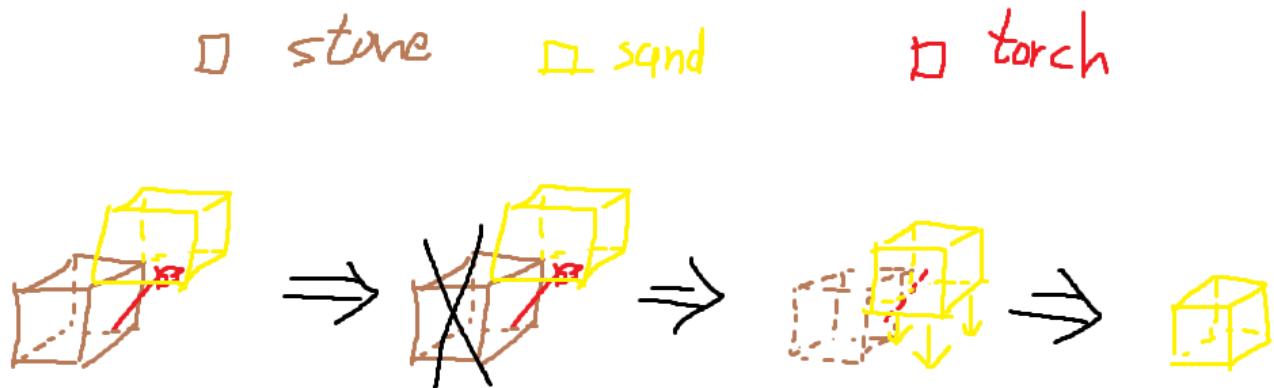


图 5: 小 M 的涂鸦

## Problem M. 晚餐

小 M 想吃丰盛的晚餐，他决定出门购买一些食材，但是他想尽可能少花力气。

小 M 的家和店铺之间的连通关系可以抽象为一颗有  $n$  个结点的树，根结点代表小 M 的家，其他结点代表店铺。每家店铺最多出售一种小 M 需要的食材，且每种需要的食材最多被一家店铺出售。小 M 可以在边上行走，耗费的力气值为（携带物品重量  $\times$  边长）。

小 M 从家开始在树上移动，经过店铺时，他可以选择在该店铺买下任意重量的食材。在此后的移动中，他必须携带这些食材；但是当经过家时，他可以将这些食材卸下，并重新回到空手的状态。

请求出小 M 购买完所有需要的食材并放回家最少需要花费的力气值。

### 输入描述

第一行一个整数  $T(1 \leq T \leq 10^5)$  表示数据组数，接下来按组给出数据。

对于每组数据：

第一行两个整数  $n(1 \leq n \leq 10^5), r$  表示树的结点个数、根结点编号（结点编号从 1 开始）。

随后  $n - 1$  行，每行两个整数  $u, v, w(1 \leq w \leq 1000)$ ，代表编号为  $u, v, w$  的结点之间有一条长度为  $w$  边。

随后一行  $n$  个整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq 1000)$ ， $a_i$  代表编号为  $i$  的结点代表的店铺中需要购买的食材的重量。 $a_r = 0$ 。

数据保证  $\sum n \leq 10^6$

### 输出描述

对于每组数据，输出一行，包含一个整数，代表最少需要花费的力气值。

### 样例

Input 1	Output 1
2	1
2 1	9
1 2 1	
0 1	
4 1	
1 2 2	
2 3 1	
3 4 1	
0 1 1 1	

### 说明/提示

第二组数据的树结构如图所示：

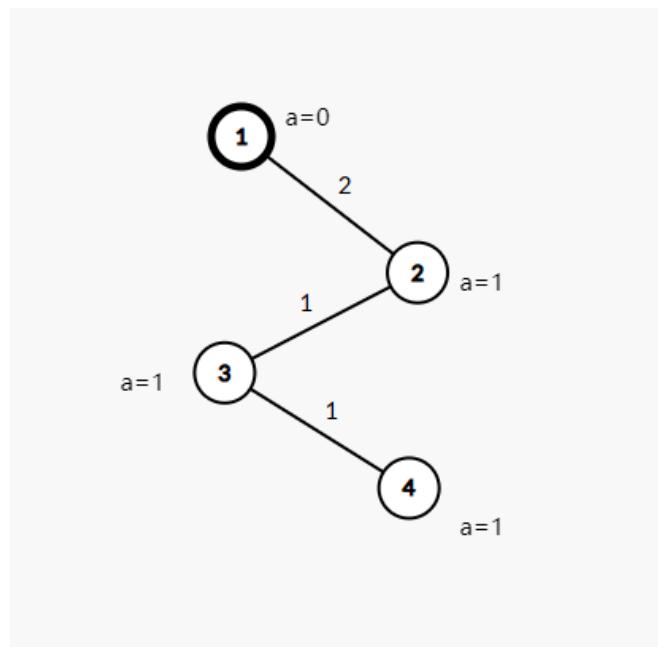


图 6: 树

小 M 可以从家里出发，空手走到结点 4。

在结点 4，购买全部的食材。携带物品重量变为 1。

向上走到结点 3，花费力气值为  $1 \times 1 = 1$ 。

在结点 3，购买全部的食材。携带物品重量变为 2。

向上走到结点 2，花费力气值为  $2 \times 1 = 2$ 。

在结点 2，购买全部的食材。携带物品重量变为 3。

向上走到结点 1，花费力气值为  $3 \times 2 = 6$ 。

放下所有食材。

总共花费的力气值为  $1+2+6=9$ 。