

城阳一中第二届 ACM 校赛

CYCPC Round 2

CYCPC 出题组

March 14, 2025

Problem List

A	秘源机兵统御械 - 疾攻
B	事已至此就赌上性命来猜谜吧
C	构造题
D	繁星坠海
E	冒泡排序
F	世界如此可爱
G	LaVI-Bavellabion
H	春日影 (MyGO!!!! ver.)
I	古明地恋学生物
J	朴素的最长路问题
K	天使的羽翼和水晶
L	笨蛋题
M	猫娘题

题目 A. 秘源机兵统御械 - 疾攻

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
内存限制: 256 megabytes



在《Genshin Impact》的新版本中，两名高人气冰系角色——主 C「柯丝克」与辅助「可爱菲」迎来了复刻。两人在技能机制上有着绝佳的相性。

然而，近日柯丝克因过度沉迷少女乐队，意外患上了"人格分裂"。这导致她的战斗方式发生了特殊的变化，需要依靠可爱菲的增伤辅助来将队伍爆发提升到极致。

柯丝克一共分裂出了 n 个人格，每个人格都拥有一个特定的技能。由于战斗设定的限制，柯丝克只能严格按照 $1, 2, \dots, n$ 的顺序依次切换人格，且每个人格在整场战斗中只能释放一次技能。

对于第 i 个人格，其技能机制如下：

- 持续输出期：技能释放后，每秒造成 a_i 点基础伤害，持续 t_i 秒。
- 技能冷却期：输出期结束后，柯丝克将立即进入持续 g_i 秒的冷却时间。

注意：在某个人格的持续输出期和技能冷却期内，柯丝克无法切换到下一个人格释放技能。

队伍中的辅助角色「可爱菲」可以释放最多 m 次强度增幅技能。该技能的机制如下：

- 增伤效果：每次增幅会提供固定持续 L 秒的增伤状态。在增伤状态下，若柯丝克原本产生的秒伤为 dam ，则实际造成的伤害将提升为 $dam \times (1 + P)$ 。
- 释放限制：可爱菲的增幅技能不可重叠（即同一时间只能存在一个增幅状态），且每次增幅的开始释放时间必须落在柯丝克当前人格的技能冷却期（即 g_i 的阶段）内。
- 技能后摇：可爱菲释放增幅技能后，会立刻进入持续 T 秒的技能后摇。在此期间，玩家处于"僵直"状态，无法进行任何操作（包括切换柯丝克的人格）。这意味着，如果后摇未结束而柯丝克的 g_i 已经结束，柯丝克也必须等待后摇结束后才能切换至下一个人格触发技能。

定义队伍的整场战斗时间为：从第 1 个人格释放技能开始，直到第 n 个人格的技能流程及可能存在的后摇完全结束为止的总时长。

定义整场战斗的 DPS（Damage Per Second，秒伤）为：

$$\text{DPS} = \frac{\text{队伍造成的总伤害}}{\text{队伍消耗的总时间}}$$

请你制定一套合理的增幅技能释放策略（包括在哪些人格的冷却期释放、在冷却期的哪个具体时间点释放），使得整场战斗的 DPS 达到最大。

求出该队伍能够达到的最大可能 DPS。

输入

第一行包含四个正整数 n, m, L, T 。

第二行包含一个小数，表示 P 。

第三到 $n + 2$ 行，每行包含三个正整数 a_i, t_i, g_i 。

输出

输出最大的可能的 DPS(Damage per second)，也就是该队伍平均每秒造成的伤害，四舍五入到整数。

样例

standard input	standard output
3 1 5 2 0.5 10 2 3 20 4 2 30 3 1	16

注释

$$1 \leq n, m \leq 1 \times 10^4$$

$$1 \leq a_i, t_i, g_i, L, T \leq 10^9$$

$$0 < P \leq 100$$

保证输入的技能增益倍率 P 最多只包含两位有效小数。

保证游戏进行的最大总时间与造成的最大总伤害，在最坏情况下的理论极限乘积不会超过 128 位有符号整数 (`__int128`) 的上限。

数据生成规则说明本题的测试数据均由官方生成器按照以下随机规则生成。令 `rand(L, R)` 表示在 $[L, R]$ 范围内均匀随机生成一个整数，`rand_float(L, R)` 表示均匀随机生成一个浮点数。给定基本参数 n (人格数量), m_{type} (次数限制类型), $mode$ (数据特性模式) 以及 max_val (数值量级上限，最高可达 10^9)。生成逻辑如下：

```
if m_type == 0: m = 0
else if m_type == 1: m = 1
else if m_type == 2: m = max(1, n / 2)
else if m_type == 3: m = n
else: m = rand(1, n)

if mode == 4:
    T = rand(1, 1000)
    L = rand(max_val / 2, max_val)
else:
    T = rand(1, max_val / 10)
    if rand(0, 1) == 0: L = rand(T, T * 5)
    else: L = rand(1, max_val / 5)

P = rand_float(0.01, 100.0)

if mode == 3:
    if rand(0, 9) == 0: a_i = rand(max_val / 2, max_val)
    else: a_i = rand(1, 10000)
else:
    a_i = rand(1, max_val)
```

```
if mode == 3:
    if a_i > max_val / 2: t_i = rand(1, 100)
    else:                  t_i = rand(1, max_val / 1000)
else:
    t_i = rand(1, max_val / 100)

if mode == 1:
    g_i = rand(T, max(T + 1000, max_val / 100))
else if mode == 2:
    if T > 1: g_i = rand(1, T - 1)
    else:    g_i = 1
else:
    g_i = rand(1, max_val / 100)
```

题目 B. 事已至此就赌上性命来猜迷吧

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

设存在 $f(n,k)$ 个有序对 (i,j) 满足 $\gcd(i,j) = k(1 \leq i < j \leq n)$ 。
给定一个正整数 $n(3 \leq n \leq 10^9)$ ，求：

$$\sum_{i=1}^n \sum_{j=1}^{\lfloor \frac{i}{2} \rfloor - 1} f(i,j)$$

结果可能很大，请求答案对 998244353 取模后的值，将取模后的答案设为 ans 。

输入

一行输入一个正整数 $T(1 \leq T \leq 100)$ ，表示测试数据个数。
接下来 T 行，每行一个正整数 n ，表示给定的正整数 n 。

输出

输出一行一个正整数，表示对于所有测试数据 ans 的异或和。

样例

standard input	standard output
3	11
3	
4	
5	

题目 C. 构造题

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

给你一段魔改的 Dijkstra 代码，构造一组节点数 ≤ 30 的数据使得这段代码内的 $push_count \geq 10^8$

```
for each vertex  $v \in G.V$ 
 $v.d = \infty$ 
 $s.d = 0$ 

 $Q =$ 
INSERT( $Q, (s, 0)$ )
 $push\_count = 0$ 

while  $Q \neq \emptyset$ 
  ( $u, d$ ) = EXTRACT-MIN( $Q$ )
  if  $d > u.d$ 
    continue
  for each vertex  $v \in G.Adj[u]$ 
    if  $u.d + w(u, v) < v.d$ 
       $v.d = u.d + w(u, v)$ 
  INSERT( $Q, (v, v.d)$ )
   $push\_count = push\_count + 1$ 
return  $push\_count$ 
```

你需要构造一张有向图，满足以下所有条件：

1. 图的节点编号为 1 到 N ，其中节点数限制为 $1 \leq N \leq 30$ 。
2. 图的有向边数限制为 $0 \leq M \leq 100$ 。
3. 边权的范围可以为负数，要求 $-10^{12} \leq w \leq 10^{12}$ 。
4. 图中绝对不能存在任何负权环（即必须保证最短路径有解且不会陷入死循环）。
5. 将你的图输入到题面的程序中（以节点 1 为起点），程序运行结束时，统计到的 $push_count$ （即向优先队列中执行插入操作的次数）必须大于等于 10^8 。

输入

本题没有输入数据。

输出

你需要输出你构造的图：

第一行输出两个正整数 N 和 M ，分别表示节点数和边数。

接下来 M 行，每行三个整数 u, v, w ，表示存在一条从节点 u 指向节点 v 的有向边，权重为 w 。（ $1 \leq u, v \leq N, u \neq v$ ）。

你可以输出重边，但不能输出自环。请确保图中不存在总和为负的环。

题目 D. 繁星坠海

输入文件: standard input
输出文件: standard output
时间限制: 5 seconds
内存限制: 1024 megabytes

辉夜姬正在乘坐飞船穿越时空寻找彩叶。

辉夜姬的飞船可以停靠在 n 个时间点。辉夜姬的飞船有 m 条航线，每条航线可以用四个整数 x, l, r, d 描述，航线的类型有两种。

- 类型 1: 此类航线从一个时刻 x 出发，可以前往在区间 $[l, r]$ 的任意一个时刻。
- 类型 2: 此类航线可以从位于区间 $[l, r]$ 的时刻出发，前往时刻 x 。

每沿着一条航线穿越一次时空，辉夜姬需要耗费一定的燃料。耗费的燃料由基础燃料 d 和附加燃油两部分组成。如果辉夜姬通过某条航线从 x 时刻前往 y 时刻，那么她必须消耗 $d + |x - y|$ 的燃料。

辉夜姬现在已经启程，出于安全考虑，她不会对 m 条航线进行更改。辉夜姬在时刻 s ，彩叶在时刻 t 。辉夜姬的飞船燃料有限，因此她希望使用最少的燃料到达彩叶身边。

请你帮帮辉夜姬！

请注意，所有航线都是单向的。

输入

第一行三个整数 n, m, s, t ，表示时刻的数量，辉夜姬所在的时刻和彩叶的时刻。

接下来 m 行，第 i 行两个整数 op_i, l_i, r_i, x_i, d_i 描述第 i 条航线。

- 如果 $op_i = 1$ ，说明该条航线为类型 1 的航线。
- 如果 $op_i = 2$ ，说明该条航线为类型 2 的航线。

输出

一行一个整数，表示消耗的最少燃料。如果无论如何辉夜姬都到达不了彩叶所在的时间点，请输出 -1 。

样例

standard input	standard output
5 5 4 1 1 3 3 2 6 1 1 5 5 1 2 2 4 1 1 2 2 3 5 9 1 1 5 4 1	4
5 5 3 1 1 1 3 1 10 2 3 4 4 1 1 2 4 1 7 1 3 5 4 10 2 1 5 4 6	-1
5 5 5 2 2 1 5 3 9 2 1 4 2 10 2 2 5 1 6 1 2 3 2 1 2 3 5 1 9	21

注释

对于所有测试点， $1 \leq m \leq 2 \times 10^5, 1 \leq n \leq 5 \times 10^5, 1 \leq l_i \leq r_i \leq n, 1 \leq x_i \leq n, 1 \leq d_i \leq 1 \times 10^{12}$ 。

题目 E. 冒泡排序

输入文件: `standard input`
输出文件: `standard output`
时间限制: 1 second
内存限制: 256 megabytes

Cubber 有一个 $n \times n$ 的多层自动化立体仓库，初始时散布着 n^2 个包裹。每个包裹都被随机赋予了一个重量 W (W 独立且等概率地从 1 到 m 的整数中生成)。

为了优化后续发货，系统需要对仓库进行一次二维的自动化整理，整理严格分为以下两个阶段：

阶段一：水平传输带整理（行排序） 在仓库的每一层（行）都有一条水平传输带。系统使用冒泡排序将每层的包裹按重量从轻到重（非降序）排列。每次在传输带上交换两个相邻包裹的位置时，只需消耗 1 个单位代价（与包裹重量无关）。

阶段二：垂直升降机整理（列排序） 水平整理完成后，仓库的每一列构成了一个垂直的升降机通道。系统再次使用冒泡排序将每一列的包裹按重量从轻到重排列（即要求轻的包裹在上，重的包裹在下）。在垂直通道中，由于需要考虑重力这一因素，交换包裹的能耗与包裹本身的重量差直接相关：每次将上方重量为 A 、下方重量为 B （且 $A > B$ ）的两个相邻包裹进行垂直对调时，机械系统需要消耗 $(A - B)$ 个单位的代价。

现请你计算整个整理过程所消耗的期望总代价。

输入

一行包含两个正整数 $n(1 \leq n \leq 10^3), m(1 \leq m \leq 10^{18})$ 。

输出

一行一个正整数，表示答案对 998244353 取模后的结果。

样例

standard input	standard output
114 514	617031371
19 19810	954576411

题目 F. 世界如此可爱

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

幻想乡需要布置一个覆盖全境的灵力结界。灵梦发现，各结界节点构成的灵力网络是一棵带权无根树，边代表灵力通路，初始权值为自然灵力。为了稳定结界，灵梦需选择一个节点作为核心，并调整通路的灵力值，使得从核心到所有叶子结界点的灵力总和相同。灵梦可以往任意通路中注入额外灵力（非负值），求她所需消耗的最小灵力总和。

简要题面：
给定一棵包含 n 个节点的带权无根树，你可以选择任意一个节点作为根，并通过在边添加非负权值，使得所有从根节点到叶子节点的路径权值和相等。求最少需要添加的权值总和。
选择 u 为根的时候叶子定义为所有度数为 1 的不为根的点。

输入

第一行包含一个整数 T ($1 \leq T \leq 1000$) 表示测试用例数。
每个测试用例的第一行包含一个正整数 n ($1 \leq n \leq 10^5$)，表示节点的个数。
接下来 $n-1$ 行，每行三个整数 a_i, b_i, w_i ($1 \leq w_i \leq 10^9$)。表示一条从 a_i 连向 b_i 权值为 w_i 的边。
所有测试用例中保证 n 的总和不超过 2×10^5 。

输出

仅包含一个整数，为最少添加的边权值。

样例

standard input	standard output
2	26
7	110
6 3 71	
2 1 59	
3 1 66	
4 2 70	
7 4 34	
5 1 84	
13	
13 10 3	
3 1 53	
5 4 83	
6 3 93	
11 2 40	
7 5 83	
4 1 34	
9 7 79	
8 2 47	
2 1 4	
10 9 61	
12 6 8	

题目 G. LaVI-Bavellabion

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

我们称一个字符串 S 与自然数 q 构成的对 (S, q) 是好的当且仅当下列条件满足:

- $0 \leq q < |S|$, 其中 $|S|$ 表示字符串 S 的长度。
- 如果执行 q 次: 将 S 中开头的字符移动至末尾。那么最终得到字符串是回文串。

我们称一个字符串 S 的价值为能与其构成好的对 q 的数量。
现在给你一个仅由小写字母构成的长度为 n 的字符串 S , 请你求出它所有子串的价值之和。

输入

第一行, 一个整数 n 表示字符串的长度。
第二行, 一个字符串 S 。

输出

一行, 一个整数, 表示好对的数量。

样例

standard input	standard output
10 abbaaacbbb	39

注释

对于所有测试点, 满足 $1 \leq n \leq 2 \times 10^5, s_i \in \{a, b, c, \dots, z\}$ 。

题目 H. 春日影 (MyGO!!!! ver.)

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

我们称一个长度为 $2m$ 的正整数序列 A 相对于 n 是好的, 当且仅当它满足:

- 对于 $1 \leq i \leq 2m$, 都满足 A_i 是 n 的因数, 即 $\frac{n}{A_i}$ 是整数。 - 令 $M = \prod_{i=1}^{2m} A_i$, 则 $M \leq n^m$ 。

现在给定 n, m , 请你求出有多少个长度为 $2m$ 的正整数序列相对于 n 是好的。

输入

一行, 两个整数, n, m 意义见题意。

输出

一行, 一个整数, 表示答案, 对 998244353 取模。

样例

standard input	standard output
7 3	42

注释

对于所有测试点, 满足 $1 \leq n \leq 10^9, 1 \leq m \leq 100$ 。

题目 I. 古明地恋学生物

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

古明地恋有 n 个小盒子，每个盒子里装了 4 张写着 1, 2, 3, 4 的卡片。初始状态下，所有盒子中的卡片顺序都是 1, 2, 3, 4。由于每个盒子有 24 种排法（即 $4! = 24$ ），桌子上的总体状态一共有 24^n 种。

现在有一个包含 m 个操作的“卡池”，分为两类：

1. 洗牌操作：共有 $m - 1$ 个。每个洗牌操作都会给出一个指令，告诉你“第 1 个盒子怎么重新排列、第 2 个盒子怎么重新排列……第 n 个盒子怎么重新排列”。题目输入给出了所有 24^n 种可能指令各自在卡池里有几个（即 c_i ）。

2. 停止操作：只有 1 个。一旦抽到它，游戏立刻结束。（所以总操作数 $m = 1 + \sum c_i$ ）

你不停地从卡池中随机抽取一个操作：

- 如果抽到“洗牌操作”：你就按照指令，把你桌子上当前的卡片状态进行一次洗牌打乱（相当于做一次长度为 4 的排列置换），然后把操作放回卡池，继续下一轮抽取。
- 如果抽到“停止操作”：游戏立刻结束，保留桌面上最终的卡片状态。

由于每次抽卡都是随机的，游戏结束时桌面上的卡片状态会有很多种可能。

请你计算：游戏结束时，桌面上变为每一种可能的排列状态的概率分别是多少？（结果需对 998244353 取模）

输入

第一行包含一个整数 n ($1 \leq n \leq 4$)，表示小盒子的个数。

第二行包含 24^n 个非负整数，第 i ($0 \leq i < 24^n$) 个数表示卡池中第 i 种“洗牌操作”的数量 c_i ($0 \leq c_i < 998244353$)。

关于第 i 种操作的定义：将整数 i 表示为 n 位 24 进制数 $\overline{p_n p_{n-1} \dots p_1}$ （不足 n 位则高位补 0）。其中 p_k ($0 \leq p_k \leq 23$) 表示该操作对第 k 个盒子的打乱指令，对应的是长度为 4 的排列中字典序第 p_k 小的排列（从 0 开始编号）。例如， $p_k = 0$ 对应排列 (1, 2, 3, 4)， $p_k = 23$ 对应排列 (4, 3, 2, 1)。

（注：停止操作的数量固定为 1，不包含在输入数据中。总操作数 $m = 1 + \sum_{i=0}^{24^n-1} c_i$ 。）

输出

为了减少输出量，你只需要输出所有答案的异或和即可。

如果存在一个答案在模 998244353 意义下不收敛，请输出 ‘-1’。

否则，输出一行一个整数，表示在计算出桌面上变成这 24^n 种状态的概率，并将这些概率对 998244353 取模后，所有 24^n 个结果的异或和（XOR sum）。

样例

standard input	standard output
1	410203005
718616568 177022934 451147478 396828834	680223012 882759996 331482250 942077878 391605507 396294

题目 J. 朴素的最长路问题

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
内存限制: 256 megabytes

有左右两排各 n 个共 $2n$ 个点，点从 1 到 $2n$ 编号，左边一排为 1 到 n ，右边一排为 $n+1$ 到 $2n$ 。

最开始有 n 条有向边，第 i 条边由 i 指向 $i+n$ ，经过这些有向边你不需要花费时间。同时有 $2n-2$ 条无向边，第 i 条连接 u_i 和 v_i ，通行需要花费 w_i 的时间。

为了让这个图尽可能美观，保证不会有一条无向边横跨左右。并且有且仅有 $n-1$ 条无向边满足 u_i, v_i 均为左边一排的节点。同时，保证左右两排排内的点相互连通。

你可以从一个点开始旅行，当你停留在点上时你可以选择通过一条有向出边或无向边走到相邻节点，当然你需要花费边所对应的时间。同时，你不会重复经过一条边，这是件无趣的事情。

接下来有 q 个问题需要你回答，每次你需要计算在从 x_i 旅游至 y_i 你所能旅行的最大时间。在点上停留的时间可以忽略不计。同时你需要保证你在任意时刻都存在一条满足要求的到达 y_i 的旅行路线。保证一定存在一条有限时间内的从 x_i 到 y_i 的路线。

输入

第一行两个正整数 n, q 。

接下来 $2n-2$ 行，每行三个正整数 u_i, v_i, w_i ，表示一条无向边。

接下来 q 行，每行两个整数 x_i, y_i 表示旅行的起点和终点。

输出

共 q 行，每行一个正整数表示能够旅行的最大时间（停留在节点上的时间可以忽略不计）。

样例

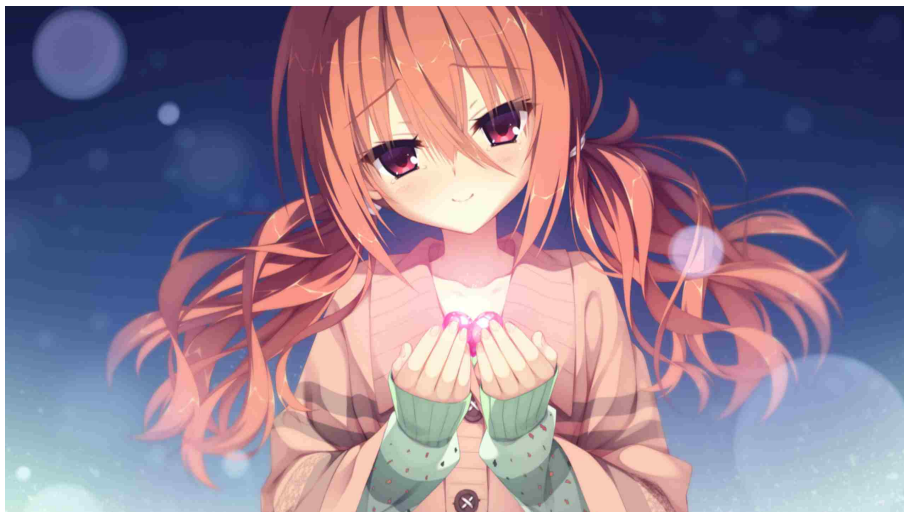
standard input	standard output
5 5	13
1 2 3	16
1 3 2	2
3 4 1	5
4 5 4	9
6 9 3	
6 10 2	
9 7 1	
8 10 2	
1 7	
5 10	
1 3	
2 3	
3 6	

注释

对于所有测试点，满足 $1 \leq n, q \leq 10^5, 1 \leq u_i, v_i, x_i, y_i \leq 2n, 1 \leq w_i \leq 10^9$ 。

题目 K. 天使的羽翼和水晶

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
内存限制: 256 megabytes



下文主要介绍网络流相关的基本知识. (摘自 OIwiki)

概述:

网络 (network) 是指一个特殊的有向图 $G = (V, E)$, 其与一般有向图的不同之处在于有容量和源汇点.

- E 中的每条边 (u, v) 都有一个被称为容量 (capacity) 的权值, 记作 $c(u, v)$. 当 $(u, v) \notin E$ 时, 可以假定 $c(u, v) = 0$.
- V 中有两个特殊的点: 源点 (source) s 和汇点 (sink) t ($s \neq t$).

对于网络 $G = (V, E)$, 流 (flow) 是一个从边集 E 到整数集或实数集的函数, 其满足以下性质.

1. 容量限制: 对于每条边, 流经该边的流量不得超过该边的容量, 即 $0 \leq f(u, v) \leq c(u, v)$;
2. 流守恒性: 除源汇点外, 任意结点 u 的净流量为 0. 其中, 我们定义 u 的净流量为 $f(u) = \sum_{x \in V} f(u, x) - \sum_{x \in V} f(x, u)$.

对于网络 $G = (V, E)$ 和其上的流 f , 我们定义 f 的流量 $|f|$ 为 s 的净流量 $f(s)$. 作为流守恒性的推论, 这也等于 t 的净流量的相反数 $-f(t)$.

对于网络 $G = (V, E)$, 如果 $\{S, T\}$ 是 V 的划分 (即 $S \cup T = V$ 且 $S \cap T = \emptyset$), 且满足 $s \in S, t \in T$, 则我们称 $\{S, T\}$ 是 G 的一个 s - t 割 (cut). 我们定义 s - t 割 $\{S, T\}$ 的容量为 $\|S, T\| = \sum_{u \in S} \sum_{v \in T} c(u, v)$.

最小割问题: 对于网络 $G = (V, E)$, 找到合适的 s - t 割 $\{S, T\}$, 使得 $\{S, T\}$ 的总容量尽可能小. 此时我们称 $\{S, T\}$ 的总容量是 G 的最小割.

给定一个包含 $n + 2$ 个节点的网络. 我们约定:

- 节点 $1, 2, \dots, n$ 为普通节点;
- 节点 $n + 1$ 为该网络的源点 (S);
- 节点 $n + 2$ 为该网络的汇点 (T).

该网络中一共包含 $3n - 1$ 条边。给定一个长度为 $3n - 1$ 的序列 a ，其中 a_k 表示第 k 条边的容量。这些边的具体构建规则如下：

1. 与源点与汇点相连的边（前 $2n$ 条边）：
对于 $1 \leq i \leq n$ ，
 - 第 $2i - 1$ 条边（奇数编号）：从源点极 S 连向普通节点 i ，其容量为 a_{2i-1} ；
 - 第 $2i$ 条边（偶数编号）：从普通节点 i 连向汇点 T ，其容量为 a_{2i} 。
2. 普通节点间的连边（后 $n - 1$ 条边）：
对于 $1 \leq i \leq n - 1$ ，
 - 第 $2n + i$ 条边：普通节点 i 连向节点 $i + 1$ ，其容量为 a_{2n+i} 。

请你求出该网络从源点 S 到汇点 T 的最小割。

输入

一行输入一个正整数 n ($1 \leq n \leq 2 \times 10^6$)。

第二行包含 $3n - 1$ 个正整数，表示容量数组 a_i ($1 \leq a_i \leq 10^9$)。

输出

输出一行一个正整数，表示答案。

样例

standard input	standard output
3 1 1 4 5 1 4 1 9	6

题目 L. 笨蛋题

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes



Cirno 和 Daiyousei 发明了一个数字博弈游戏。

游戏初始时有一个正整数 n 。两人轮流进行操作，由 Cirno 先手。在每个回合中，当前玩家必须选择当前数字 n 的一个严格约数 d （这意味着 d 必须能够整除 n ，且 $1 \leq d < n$ ），然后将 n 减去 d ，即令 $n = n - d$ 。

当某个玩家面对数字 $n = 1$ 时，由于找不到满足要求的约数 d （因为不允许 $d = 1$ 且 $d < 1$ ），该玩家将无法进行操作，从而立刻输掉游戏。

Cirno 和 Daiyousei 都绝顶聪明，每次都会采取最优策略。给定初始数字 n ，请问谁将赢得这场游戏？

输入

第一行包含一个整数 t ($1 \leq t \leq 10^4$)，表示测试用例的数量。

接下来的 t 行，每行包含一个整数 n ($1 \leq n \leq 10^{18}$)，表示游戏开始时的初始数字。

输出

对于每个测试用例，输出一行。如果是 Cirno 获胜，输出 Cirno；如果是 Daiyousei 获胜，输出 Daiyousei。

样例

standard input	standard output
4	Cirno
2	Daiyousei
3	Cirno
4	Daiyousei
7	

题目 M. 猫娘题

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
内存限制: 256 megabytes

在探索一个古老的二维网格迷宫时，发现了许多被困住的猫娘。这些猫娘被一种名为"折线箭头"的魔法结界封印在网格的整数坐标点上。为了拯救她们，冒险者需要解除这些结界。

结界的规则非常奇怪：每个猫娘都面向一个特定的方向（共 8 个方向）。如果一个猫娘所面对方向的射线上没有其他猫娘阻挡，她就可以被安全地移出地图（解除封印）。一旦一个猫娘被移出，她原本占据的位置就会变空，不再阻挡其他猫娘的视线。

冒险者当时救出了所有猫娘，成为了传说中的英雄。

如今，冒险者再次深入古老迷宫，发现猫娘们经过魔法强化后，每只都拥有了完整的身子！她们不再是孤零零的一个点，而是一个由多个相邻网格细胞组成的 4 连通块。冒险者必须更加小心——现在任何猫娘的身子细胞都会阻挡其他猫娘的视线。

只有当一只猫娘眼睛位置发出的射线上（不含自身）没有任何其他猫娘的任何身子细胞时，她才能被安全救出。一旦救出，她整个身子的所有细胞会瞬间消失，不再阻挡任何人。

冒险者希望以字典序最小的顺序救出所有猫娘。如果存在多种可行顺序，请输出字典序最小的那个；若无法全部救出，请输出 -1 。

有 N 只猫娘。每只猫娘 i 包含：

- 眼睛位置 (ex_i, ey_i)
- 面向方向 d_i
 - 0 北 (North)
 - 1 东北 (North-east)
 - 2 东 (East)
 - 3 东南 (South-east)
 - 4 南 (South)
 - 5 西南 (South-west)
 - 6 西 (West)
 - 7 西北 (North-west)
- 一个连通的身子，由 K_i 个互不相同的网格细胞组成（保证包含眼睛位置，且 4 连通）

移除规则：

对于尚未被移除的猫娘 i ，若从她的眼睛位置 (ex_i, ey_i) 沿方向 d_i 发出的无限射线（不含起点）上不存在任何其他猫娘的身子细胞，则她可以被移除。

移除后，该猫娘所有 K_i 个细胞同时变空。

请判断是否能救出全部猫娘：

1. 若能，输出字典序最小的合法移除顺序（猫娘编号 $1 \sim N$ 的排列）。
2. 若不能，输出 -1 。

输入

第一行包含一个正整数 N 。

接下来为 N 只猫娘的输入，每只格式如下：

- 第一行: $ex_i ey_i d_i K_i$
- 接下来 K_i 行: 每行两个整数 $x y$, 表示该猫娘占据的一个细胞坐标。

输出

若无法全部移除, 输出一行 -1 。

若可行, 输出一行 N 个整数 ($1 \sim N$ 的排列), 表示按顺序移除的猫娘编号, 请输出字典序最小的一种方案。

样例

standard input	standard output
2 0 0 2 1 0 0 1 0 6 2 1 0 2 0	-1
3 0 0 2 3 0 0 1 0 2 0 0 2 4 1 0 2 3 0 4 2 3 0 4 0	3 1 2

注释

- $1 \leq N \leq 2 \times 10^3$
- $1 \leq K_i \leq 10^3$
- 保证 (ex_i, ey_i) 出现在该猫娘的 K_i 个细胞中。
- 该猫娘的 K_i 个细胞 4 连通 (上下左右相邻)。
- 所有猫娘的细胞全局不重叠 (没有两个猫娘共用同一格子)。
- $|x|, |y| \leq 10^9$
- $0 \leq d_i \leq 7$
- 猫娘编号为输入顺序 ($1 \sim N$)。