



Problem A

超维攻坚

Time limit: 4 seconds

比特游戏公司最近发布了一款新游戏《超维攻坚》。超维攻坚战是该游戏的最终 BOSS 战， n 名玩家组成攻坚队对外宇宙的强大存在进行讨伐。所有玩家可以看作一个个三维空间内的质点，其中第 i 个玩家的坐标是 (x_i, y_i, z_i) 。

外宇宙的强大存在发动了它的大招：首先它控制住了所有玩家，使得所有玩家被固定在原位置不能移动；此外由于它是超维生物，因此它的每次攻击投射到三维可以视作选定实数 a, b, c, d ，并秒杀所有满足 $ax + by + cz + d \geq 0$ 的玩家。外宇宙的强大存在可以进行任意多次攻击，但每次攻击不一定会瞄准玩家。

BOSS 大招演出完毕后，已知幸存了至少一名玩家，请问有多少种可能的局面？两个局面被认为不同当且仅当存在一名玩家在一个局面中幸存，却在另一个局面中阵亡。

Input

第一行包含一个正整数 T ($1 \leq T \leq 10$)，表示测试数据的组数。

每组数据第一行包含一个正整数 n ($1 \leq n \leq 15$)，表示攻坚队中玩家的数量。

接下来 n 行，每行三个整数 x_i, y_i, z_i ($0 \leq x_i, y_i, z_i \leq 100$)，依次表示每个玩家的坐标。输入数据保证玩家所处的坐标两两不同。

Output

对于每组数据输出一行一个整数，即大招演出完毕后可能的局面数量。

Sample Input	Sample Output
1 5 0 0 0 4 0 0 2 4 0 2 2 5 2 2 1	30



Problem B

黑白边游戏

Time limit: 2.5 seconds

小 Q 和小 T 在一张 n 个点的完全无向图上玩游戏。在游戏的一开始，图中所有边都是黑色。他们会轮流操作 m ($m \bmod 2 = 0$) 轮，小 Q 先手，即小 Q 将操作第 $1, 3, 5, \dots, m-1$ 轮，小 T 将操作第 $2, 4, 6, \dots, m$ 轮。每轮的操作方需要在图中选择恰好一条边，将其颜色进行反转，即由黑变白或者由白变黑，然后操作方将获得 $\sum_{i=1}^n \sum_{j=1}^n dis(i, j)$ 点得分，其中 $dis(i, j)$ 表示点 i 到点 j 的最短路径的长度。在所有 m 轮操作结束之后，谁总分高谁就获胜。

在第 i 轮中，每条黑边的长度都为 a_i ，每条白边的长度都为 b_i 。小 Q 和小 T 双方都知道所有 m 轮的 a, b 数据，且他们都会以最优策略进行操作，目标是让自己的总分减去对手的总分尽可能大。作为观战方的你，能否写一个程序预测最后双方的分差？

Input

第一行包含一个正整数 T ($1 \leq T \leq 50$)，表示测试数据的组数。

每组数据第一行包含两个正整数 n, m ($2 \leq n \leq 8, 2 \leq m \leq 300, m \bmod 2 = 0$)，分别表示图中的点数以及游戏的轮数。

接下来 m 行，每行两个正整数 a_i, b_i ($1 \leq a_i, b_i \leq 5$)，依次表示每轮中黑边和白边的长度。

输入数据保证 $\sum m \leq 2000$ 。

Output

对于每组数据输出一行一个整数，即最后小Q的总分减去小T的总分的值。

Sample Input	Sample Output
2	0
2 4	-5 6
1 2	
3 1	
5 3	
2 5	
5 4	
1 1	
2 2	
3 5	
5 2	



Problem C

最优 K 子段

Time limit: 6 seconds

给定一个序列 a_1, a_2, \dots, a_n ，从中找出恰好 k 个不相交连续子段，满足每一段的长度都是质数。假设其中第 i 个子段的子段和是 s_i ，你需要最大化 $\min\{s_1, s_2, \dots, s_k\}$ 。

Input

第一行包含一个正整数 T ($1 \leq T \leq 100$)，表示测试数据的组数。

每组数据第一行包含两个正整数 n, k ($2 \leq n \leq 200\,000, 1 \leq k \leq n$)。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($|a_i| \leq 1000$)。

输入数据保证 $\sum n \leq 10^6$ ，且每个 a_i 都是在 $[-1000, 1000]$ 均匀随机生成得到（样例除外）。

Output

对于每组数据输出一行：若存在合法方案，输出一个整数，即 $\min\{s_1, s_2, \dots, s_k\}$ 的最大可能值；若无解，输出 “impossible”。

Sample Input

```
4
6 1
1 1 4 5 1 4
6 1
-1 -1 -4 -5 -1 -4
6 3
-1 -1 -4 -5 -1 -4
2 2
0 0
```

Sample Output

```
15
-2
-9
impossible
```



Problem D

分组

Time limit: 10 seconds

给定 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i < 2^m$) 以及 0 到 $2^m - 1$ 的权重 $w_0, w_1, \dots, w_{2^m-1}$; 你需要把这 n 个正整数分成四组 A, B, C, D , 令 $f(A), f(B), f(C), f(D)$ 分别表示每组中所有数字的异或和, 你的分组方案需要最小化 $w_{f(A)}, w_{f(B)}, w_{f(C)}, w_{f(D)}$ 的极差, 即:

$$\max\{w_{f(A)}, w_{f(B)}, w_{f(C)}, w_{f(D)}\} - \min\{w_{f(A)}, w_{f(B)}, w_{f(C)}, w_{f(D)}\}$$

注意: 每组都可以为空, 此时 $f(\cdot) = 0$ 。

Input

第一行包含一个正整数 T ($1 \leq T \leq 5$), 表示测试数据的组数。

每组数据第一行包含两个正整数 n, m ($4 \leq n \leq 18, 1 \leq m \leq 10$)。

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i < 2^m$)。

第三行包含 2^m 个整数 $w_0, w_1, \dots, w_{2^m-1}$ ($0 \leq w_i \leq 10^9$)。

Output

对于每组数据输出一行一个整数, 即 $w_{f(A)}, w_{f(B)}, w_{f(C)}, w_{f(D)}$ 的极差的最小可能值。

Sample Input	Sample Output
2 4 2 1 2 3 1 0 1 2 3 7 4 3 2 15 13 11 9 7 12 13 14 15 16 1 2 3 4 5 6 7 8 9 10 11	1 2



Problem E

多层血条

Time limit: 1 second

在本题中你将帮助比特公司的新游戏《超维攻坚》实现 BOSS 战的多层血条功能。血条由 n 行 m 列个像素构成，外围还有一圈 1 像素厚的边框，样式请参照样例。

假设 BOSS 有 k 条血，则血条由 $k + 1$ 个图层自底向上叠加而成。底层是 n 行 m 列个空格，上面的 k 层自底向上依次表示第 $1, 2, \dots, k$ 条血。每条血都有 n 行，每一列代表 1 点生命值，因此一共有 $k = \lceil \frac{hp}{m} \rceil$ 条血。每条血都由单一字符构成，自底向上由字符 ‘A’ 到 ‘E’ 循环表

示，例如：第一条血为 ‘A’，第二条血为 ‘B’，第五条血为 ‘E’，第六条血为 ‘A’。

除此之外，为了加强击打反馈，假设 BOSS 有 hp 点生命值，玩家当前的攻击对 BOSS 造成了 dmg 点伤害，那么第 $hp - dmg + 1$ 至第 hp 点生命值代表的像素将被替换成字符 ‘.’。

给定 n, m, hp, dmg ，请绘制出对应的多层血条。

Input

第一行包含一个正整数 T ($1 \leq T \leq 50$)，表示测试数据的组数。

每组数据包含一行四个整数 n, m, hp, dmg ($1 \leq n \leq 50, 5 \leq m \leq 800, 1 \leq hp \leq 10^9, 0 \leq dmg < hp$)，分别表示血条的尺寸、BOSS 的生命值以及当前受到的伤害。

Output

对于每组数据输出 $n + 2$ 行，每列 $m + 2$ 个字符，即对应的多层血条。



Sample Input

```
5
2 5 4 2
1 10 49 1
1 10 52 0
1 10 52 5
1 10 52 50
```

Sample Output

```
+-----+
|AA.. |
|AA.. |
+-----+
+-----+
|EEEEEEEE.D|
+-----+
+-----+
|AAAAAAAAEE|
+-----+
+-----+
|..EEEEEE..|
+-----+
+-----+
|.....|
+-----+
```



Problem F

延时操控

Time limit: 4 seconds

给定一张 n 行 n 列的网格地图，其中 ‘#’ 表示障碍，剩下的字符表示空地。地图上有恰好一名己方角色 ‘P’ 和敌方角色 ‘E’。你需要帮助小 Q 操控己方角色去击败敌方角色成功通关游戏。在每一回合中，你需要键入恰好一个字符以表示操作指令，分为以下四种：

- ‘L’：从 (x, y) 移动至 $(x, y - 1)$ 。
- ‘R’：从 (x, y) 移动至 $(x, y + 1)$ 。
- ‘U’：从 (x, y) 移动至 $(x - 1, y)$ 。
- ‘D’：从 (x, y) 移动至 $(x + 1, y)$ 。

如果目的地是障碍或者不在地图内，那么本次移动将会失败。己方角色可以与敌方角色重叠，也可以穿透对方。一旦移动失败，移动方将扣除 1 点生命值，并在这一回合待在原地。由于小 Q 选的是最高难度，因此己方没有任何容错率，而敌方有 hp 点生命值。敌方不会自主移动，只会复读己方 k 回合之前的指令，你需要善用该机制来“延时操控”敌方以获得胜利。

游戏即将从第 1 回合开始，一共执行 m 个回合；第 i 回合的执行逻辑如下：

1. 键入第 i 回合的操作指令 c_i 。
2. 尝试按照指令 c_i 移动己方角色，若移动失败则游戏立即结束。
3. （若 $i \leq k$ 则忽略本条）尝试按照指令 c_{i-k} 移动敌方角色，若移动失败则敌方角色扣除 1 点生命值，若生命值为 0 则游戏立即通关。

请写一个程序，统计有多少种可能的操作序列使得你可以通关游戏。两个操作序列被认为不同当且仅当长度不同或者某一回合的指令不同。请注意：如果第 m 回合仍然没有将敌方的生命值归零，则这样的方案不满足条件。

Input

第一行包含一个正整数 T ($1 \leq T \leq 100$)，表示测试数据的组数。

每组数据第一行包含四个整数 n, m, k, hp ($2 \leq n \leq 50, 1 \leq m \leq 50, 0 \leq k < m, 1 \leq hp \leq 5$)，分别表示地图的尺寸、总回合数、延时回合数以及敌方的生命值。

接下来 n 行，每行一个长度为 n 的字符串，由 ‘#’、‘.’、‘P’、‘E’ 构成，描述地图。输入数据保证恰好存在一个 ‘P’ 和一个 ‘E’。

输入数据保证最多只有 10 组数据满足 $n > 20$ 。



Output

对于每组数据输出一行一个整数，即能通关的操作序列数量，由于答案可能很大，请对 $10^9 + 7$ 取模输出。

Sample Input	Sample Output
2 2 3 0 2 . # PE 5 5 2 1P.E...	1 124

Hint

对于第一组样例，满足条件的操作序列只有一组：“UD”。



Problem G

序列更新

Time limit: 6 seconds

给定两个长度为 n 的序列 a_0, a_1, \dots, a_{n-1} 和 b_0, b_1, \dots, b_{n-1} ，你需要依次执行 q 次操作，每次操作将会给出一个整数 k ($0 \leq k < n$)，对于每个 i ($0 \leq i < n$)，你需要将 a_i 更新为 $\max(a_i, b_{(i+k) \bmod n})$ 。为了证明你确实维护了 a 序列，请在每次操作之后输出 $\sum_{i=0}^{n-1} a_i$ 的值。

Input

第一行包含一个正整数 T ($1 \leq T \leq 2$)，表示测试数据的组数。

每组数据第一行包含两个正整数 n, q ($1 \leq n, q \leq 250\,000$)，分别表示序列的长度以及操作的次数。

第二行包含 n 个正整数 a_0, a_1, \dots, a_{n-1} ($1 \leq a_i \leq 10^9$)。

第三行包含 n 个正整数 b_0, b_1, \dots, b_{n-1} ($1 \leq b_i \leq 10^9$)。

接下来 q 行，每行一个整数 k ($0 \leq k < n$)，依次描述每次操作。

输入数据保证每个 a_i, b_i 都是在 $[1, 10^9]$ 均匀随机生成得到（样例除外），且每个 k 都是在 $[0, n)$ 均匀随机生成得到（样例除外）。

Output

对于每组数据输出 q 行，其中第 i 行输出一个整数，即在第 i 次操作完毕之后 $\sum_{i=0}^{n-1} a_i$ 的值。

Sample Input

Sample Output

1	29
5 5	31
1 5 3 6 8	33
2 5 4 7 3	35
3	36
2	
4	
1	
0	



Problem H

魔法卡牌

Time limit: 3 seconds

Alice 有 n 张卡牌，第 i ($1 \leq i \leq n$) 张卡牌的正面有数字 a_i ，背面有数字 b_i ，初始时所有卡牌正面朝上。

现在 Alice 可以将任意张（包括 0 张或者全部 n 张）卡牌翻面，即由正面朝上改为背面朝上。这 n 张卡牌是有魔法的，Alice 必须遵守全部 m 条规则，每条规则是下面两种之一：

- “A \times y” ($1 \leq x, y \leq n, x \neq y$): 仅考虑最终的局面而不考虑过程，如果最终第 x 张卡牌正面向上，那么最终第 y 张卡牌必须背面向上。
- “B \times y” ($1 \leq x, y \leq n, x \neq y$): 仅考虑最终的局面而不考虑过程，如果最终第 x 张卡牌正面向上，那么最终第 y 张卡牌必须正面向上。

Alice 的目标是让最终朝上的 n 个数字的总和尽量大。请你帮 Alice 算一算总和的最大值是多少。

Input

第一行包含一个正整数 T ($1 \leq T \leq 300$)，表示测试数据的组数。

每组数据第一行包含两个正整数 n, m ($2 \leq n \leq 60, 1 \leq m \leq 5000$)，分别表示卡牌和规则的数量。

接下来 n 行，每行两个整数 a_i, b_i ($0 \leq a_i, b_i \leq 10^7$)，依次描述每张卡牌正面和背面的数字。

接下来 m 行，每行描述一条规则。

输入数据保证最多只有 10 组数据满足 $n > 30$ 。

Output

对于每组数据输出一行两个整数：最终朝上的 n 个数字的总和的最大值以及对应的方案数。两个方案被认为不同当且仅当存在一张卡牌在一个方案中正面向上，并在另一个方案中背面向上。由于一定可以将所有卡牌背面向上，因此一定存在合法方案。



Sample Input

```
1
3 2
5 3
1 3
3 1
A 1 3
B 1 2
```

Sample Output

```
9 1
```



Problem I

昵称检索

Time limit: 3 seconds

给定一个字符串 S ，请计算从中删去一部分字符后（可以什么都不删），可以得到多少个本质不同的“昵称”。

一个字符串被认为是“昵称”当且仅当它可以被划分成前后两个部分，第一个部分是给定的 n 个名字之一，第二个部分长度固定四位，表示生日。

例如第一个样例中，可以得到的“昵称”分别为：“kevin0724”、“kevin0729”、“kevin0924”、“kevin0929”。

Input

第一行包含一个正整数 T ($1 \leq T \leq 100$)，表示测试数据的组数。

每组数据第一行包含两个正整数 n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 10^6$)，分别表示名字的数量以及字符串 S 的长度。

第二行包含一个长度为 m 的字符串 S ，由数字和小写英文字母构成。

接下来 n 行，每行一个长度在 $[1, 20]$ 之间的仅由小写英文字母构成的字符串 $name_i$ ，表示一个名字。输入数据保证名字两两不同。

输入数据保证 $\sum m \leq 7\,000\,000$ ，且 $\sum |name_i| \leq 7\,000\,000$ 。

Output

对于每组数据输出一行一个整数，即能得到的本质不同的“昵称”数量。

Sample Input

```
2
1 18
k9e9v9i9n909792949
kevin
2 24
alicealicebobbob02290229
alice
bob
```

Sample Output

```
4
18
```



Problem J

矩阵的周期

Time limit: 1.5 seconds

给定一个 $n \times n$ 的 01 矩阵 M , 令 $f(i, j, k) = [(M^k)_{i,j} \neq 0]$, 请对于每对 (i, j) ($1 \leq i, j \leq n$), 找出最小的正整数 t , 满足当 k 充分大时必有 $f(i, j, k) = f(i, j, k + t)$ 。

Input

第一行包含一个正整数 T ($1 \leq T \leq 20$), 表示测试数据的组数。

每组数据第一行包含一个正整数 n ($1 \leq n \leq 60$), 表示矩阵的大小。

接下来 n 行, 每行一个长度为 n 的 01 串, 第 i 行第 j 列表示 $M_{i,j}$ ($1 \leq i, j \leq n$)。

Output

对于每组数据输出 n 行, 第 i ($1 \leq i \leq n$) 行输出 n 个整数, 其中第 j ($1 \leq j \leq n$) 个整数表示最小的正整数 t , 满足当 k 充分大时必有 $f(i, j, k) = f(i, j, k + t)$; 若找不到这样的 t , 输出 “-1”。

Sample Input

```
1
9
010010000
001000001
000100000
010000000
000001000
000000100
000000010
000010001
000000000
```

Sample Output

```
1 3 3 3 4 4 4 4 12
1 3 3 3 1 1 1 1 3
1 3 3 3 1 1 1 1 3
1 3 3 3 1 1 1 1 3
1 1 1 1 4 4 4 4 4
1 1 1 1 4 4 4 4 4
1 1 1 1 4 4 4 4 4
1 1 1 1 4 4 4 4 4
1 1 1 1 1 1 1 1 1
```



Problem K

找环

Time limit: 12 seconds

给定一张 n 个点 m 条边的有向图，可能有重边，也可能有自环。点的编号依次为 1 到 n ；第 i 条边 $u_i \rightarrow v_i$ 的长度为 $998\,244\,353^{w_i}$ 。

请写一个程序，在给定的图中找到一个环，使得环上所有边的平均长度最小，或判断无解。

Input

第一行包含一个正整数 T ($1 \leq T \leq 100$)，表示测试数据的组数。

每组数据第一行包含两个正整数 n, m ($1 \leq n \leq 1000, 1 \leq m \leq 2000$)，分别表示点数和边数。

接下来 m 行，每行三个整数 u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, 0 \leq w_i < n$)，表示边 $u_i \rightarrow v_i$ 的长度为 $998\,244\,353^{w_i}$ 。

输入数据保证最多只有 10 组数据满足 $n > 100$ 或 $m > 100$ 。

Output

对于每组数据输出一行：若找不到环，输出 “-1”，否则假设答案是 $\frac{p}{q}$ ，你需要输出最小的非负整数 r 满足 $q \cdot r \equiv p \pmod{(10^9 + 7)}$ 。你可以认为这样的 r 一定存在。



Sample Input

Sample Output

6	1
3 3	-1
1 2 0	499122177
2 3 0	499122177
3 1 0	540815376
2 1	1
1 2 0	
2 2	
1 2 0	
2 1 1	
3 5	
1 2 0	
1 3 1	
2 3 2	
3 2 0	
2 3 1	
4 5	
1 2 3	
2 3 3	
3 1 3	
2 4 1	
4 1 1	
1 1	
1 1 0	



Problem L

寻找宝藏

Time limit: 6 seconds

你得到了一张藏宝图，这上面标记了埋藏在地底下的 n 个海盗藏宝箱，编号依次为 1 到 n ，第 i 个宝箱的坐标是 (i, p_i) ，打开它你将得到 v_i 枚金币。

你现在位于 $(0, 0)$ ，每次你可以选择从 (x, y) 移动到 $(x + 1, y)$ 或者 $(x, y + 1)$ ，当你位于某个宝箱正上方时，你将可以挖出它并拿走里面的所有金币。

不幸的是，有一个危险的陷阱区域没有被标记出来！通过多方调研，你得知这是一个边平行坐标轴的矩形区域，它是 m 种可能的位置分布之一。请对于每种可能的情况分别计算按照最优路线你能拿走多少金币。

假设陷阱区域的位置分布是第 i 种可能，假设它是以 (x_1, y_1) 和 (x_2, y_2) 为对顶点的矩形，那么 (x, y) 是陷阱当且仅当 $x_1 \leq x \leq x_2$ 且 $y_1 \leq y \leq y_2$ 。你的路线不能途径任何陷阱点。当然，你只需要考虑当前的第 i 个矩形，不需要考虑其它 $m - 1$ 个矩形。

Input

第一行包含一个正整数 T ($1 \leq T \leq 100$)，表示测试数据的组数。

每组数据第一行包含两个正整数 n, m ($1 \leq n, m \leq 300\,000$)，分别表示宝箱的数量以及可能的矩形数。

接下来 n 行，第 i 行包含两个正整数 p_i, v_i ($1 \leq p_i \leq n, 1 \leq v_i \leq 10^9$)，依次描述每个宝箱。输入数据保证 p_i 互不相同。

接下来 m 行，每行四个正整数 x_1, y_1, x_2, y_2 ($1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq n$)，依次描述每种可能的矩形陷阱区域。

输入数据保证 $\sum n \leq 1\,500\,000$ ，且 $\sum m \leq 1\,500\,000$ 。

Output

对于每组数据输出 m 行，其中第 i 行输出一个整数，即在危险陷阱区域是第 i 个矩形的情况下你最多能拿走的金币数量。



Sample Input

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

Sample Output

```
7
5
4
3
0
```