

注意事项

- 文件名（程序名和输入输出文件名）必须使用英文小写字母。
- 选手只需提交一个文件夹作为选手目录，代码文件直接放在选手目录内。
- 由于一些原因（指出题人太懒了），本套题目仅支持语言 C/C++。
- 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。
- 评测环境为 Windows 10 64-bit，使用 LemonLime v0.2.2 进行评测。
- 编译器版本为 MinGW-W64 8.1.0 64-bit，支持 C++11，开启 O2 优化，栈空间与题目空间限制相同。
- 所有题目的时间与空间限制均不少于标程在相同评测环境下所消耗的 2 倍。
- 对于存在子任务的题目，子任务 A 依赖子任务 B，当且仅当任何满足子任务 B 的限制的数据，都能满足子任务 A 的限制。
- **请注意，题目难度与题目顺序无关。**
- 题目很简单，部分分很良心。如果你见过原题或者没见过原题但 AK 了，请不要 D 出题人。

Problem A. 赛博朋克 2077 (cyberpunk.c/cpp)

时间限制：1s

空间限制：512MB

早上好，夜之城！昨天的死人乐透，最后结果是满打满算的整整三十个！

多亏了没完没了的帮派火并，光海伍德就挂了十个。但有一位警官也挂了，我看你们全得赔钱，因为 NCPD 肯定咽不下这口气。

圣多明戈再次发生停电。电网遭人破坏，又是网络黑客干的好事。

与此同时，威斯特布鲁克的创伤小组还在人行道上替赛博精神病的受害者收尸。

而在太平洋呢？好吧……太平洋还是那个太平洋。

我是你们的铁哥们斯坦。和我一起，开始逐梦之城的新一天吧！

小 S 最近在玩一款叫做《赛博朋克 2077》的游戏。在这款游戏中，小 S 扮演的角色，V，为了替好友杰克、T-BUG 和强尼·银手复仇，为了找到取出体内生物芯片的方法，为了追寻在这个迷乱疯狂的赛博朋克世界中生命的真正价值，决定独自闯入商业巨头“荒坂公司”的总部，荒坂塔。

V 并非有勇无谋之辈，她首先要做的是攻破荒坂塔中的人工智能防御网络。在夜之城过去最强大的黑客，奥特·坎宁安的帮助下，她成功获取了人工智能防御网络的最高权限。她发现，防御网络的形态是一棵深度为 h 的完全二叉树（每个非叶节点都恰好有两个子节点，共有 $2^h - 1$ 个节点），且在时刻 0，她控制了二叉树的根节点。在接下来的每个时刻，V 都可以选择至多 p 个她控制了节点，并将它们同时瘫痪。任何节点被瘫痪时，如果它存在子节点，那么它的子节点将立刻被 V 控制。

留给 V 的时间不多了。她想知道她最短能用多少时间来瘫痪所有节点。

小 S 很喜欢这款游戏，所以她玩了很多个周目，而每个周目的 h 和 p 都是不同的，你需要对所有 t 个周目的情况做出回答。

Input

输入从文件 `cyberpunk.in` 中读取。

第一行一个正整数 t ，代表小 S 重复游玩的周目数。

接下来 t 行，每行两个正整数 h 和 p ，分别代表某个周目中，二叉树的深度和 V 每个时刻能选择的节点个数。

Output

输出到文件 `cyberpunk.out` 中。

输出 t 行，每行一个正整数，代表在这个周目中， V 瘫痪所有节点的最短用时。

Sample Input 1

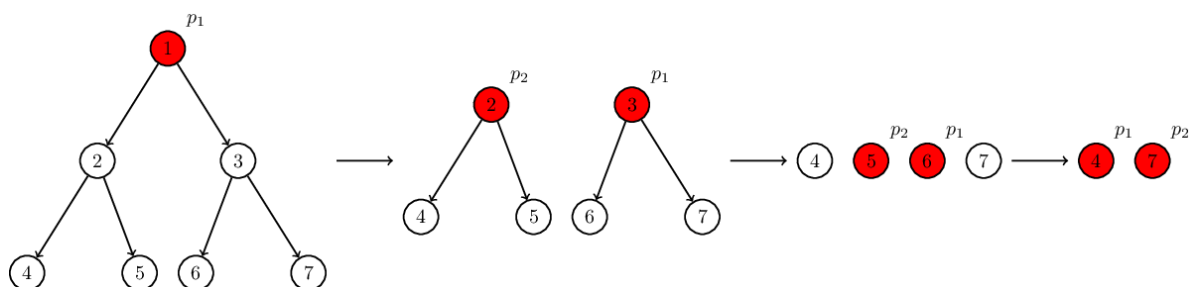
```
3
3 1
3 2
10 6
```

Sample Output 1

```
7
4
173
```

Sample Explanation 1

这张图解释了样例中的第二组数据，每个时刻 V 选择瘫痪的节点已用红色标出。



Subtasks & Constraints

本题采用子任务制评测，你能得到该子任务的分数当且仅当你通过了这个子任务中的**所有**测试点。对所有的数据，保证 $1 \leq t \leq 5 \times 10^5, 1 \leq h \leq 50, 1 \leq p \leq 10^9$ 。

- Subtask 1 (30) : $h \leq 10$ 。
- Subtask 2 (30) : $p \leq 10^4$ 。
- Subtask 3 (40) : 无特殊限制。

Problem B. 美味星球 (planet.c/cpp)

时间限制：2s

空间限制：512MB

“这个是灰球君.....它能自动吃掉卫生间里的所有灰尘！”

.....

灰球君 把所有东西都吃掉了！它的质量变得过于巨大.....于是只能爆炸！

于是，宇宙产生了.....

小 W 最近在玩一款叫做《美味星球》(Tasty Planet) 的游戏。在这款游戏中, 小 W 需要操纵一只灰球君, 吃掉所有比它小的物体来逐渐长大, 直至吞掉整个宇宙.....

现在, 灰球君 来到了一栋无人的房屋中, 它由 n 个房间和 m 条连接这些房间的无向走廊组成。灰球君 知道, 每个房间内恰好都有一份食物(当然, 灰球君 不止能吃掉食物.....), 吃掉第 i 个房间内的食物会让灰球君 的直径增加 c_i 个单位。这里我们认为, 不管灰球君的直径是多少, 它都一定能吃掉食物。尽管房间足够大, 但这些走廊却十分狭窄。具体来说, 第 i 条走廊连接了第 a_i 个和第 b_i 个房间, 且它有一个宽度 w_i 。如果灰球君 的直径**大于**走廊的宽度, 它就无法通过这条走廊了。

灰球君 十分聪明, 它懂得忍耐, 因此可以在进入某个房间后不直接吃掉这个房间内的食物; 但同时灰球君 也十分饥饿, 因此它想要吃掉所有 n 份食物。开始时, 灰球君 在 1 号房间中, 但它可以在吃完 n 份食物后在任意的房间离开。灰球君 不能在房屋中缩小自己的直径, 但它可以在进入房屋前任意调整。它调整直径的方法是先把自己变得充分大, 再不停地撞击一块坚硬的石头来减少自身物质的量, 因此它不希望自己的直径被缩减得过小。现在, 灰球君 想要知道, 它能否在以一个**正整数**直径进入房屋时吃完所有食物; 如果可以, 它还希望知道它最大能以什么直径进入房屋。

Input

输入从文件 `planet.in` 中读取。

第一行两个正整数 n, m , 分别表示房间和无向走廊的数量。

第二行 n 个正整数 c_1, c_2, \dots, c_n , 第 i 个正整数表示吃掉第 i 个房间的食物后会让灰球君 的直径增加的单位数。

接下来的 m 行, 第 i 行包含三个正整数 a_i, b_i, w_i , 分别表示第 i 条走廊两端的房间编号和这条走廊的宽度。保证任意两个房间均由走廊直接或间接相连, 保证无自环, 但**不保证**无重边。

Output

输出到文件 `planet.out` 中。

第一行一个整数, 表示想要吃掉所有食物, 灰球君 最大可以以什么直径进入房屋。如果无论以什么**正整数**直径进入房屋都无法吃完所有食物, 输出 -1 。

Sample Input 1

```
3 3
1 2 3
1 2 4
1 3 4
2 3 6
```

Sample Output 1

```
3
```

Sample Explanation 1

如果以直径 3 进入房屋, 可以沿着路径 $1 \rightarrow 2 \rightarrow 3$ 行进, 同时吃掉所有沿途的食物。假如以直径 4 或更大进入房屋, 可以发现无论如何都无法在吃掉房间 1 中的食物后进入房间 2 或 3, 也无论如何都不可能在吃掉房间 2 和 3 中的食物后进入房间 1。

Sample Input 2

```
2 1
1 1
1 2 1
```

Sample Output 2

```
-1
```

Sample Explanation 2

显然，即使以最小的正整数直径 1 进入房屋，也无法吃完所有食物。

Sample Input/Output 3

见选手下发文件中 `ex_planet3.in/ex_planet3.ans`。

Subtasks & Constraints

本题采用子任务制评测，你能得到该子任务的分数当且仅当你通过了这个子任务中的**所有**测试点。对所有的数据，保证 $2 \leq n \leq 10^5, n-1 \leq m \leq 2 \times 10^5$ 。对任意的 $i \in [1, m] \cup \mathbb{Z}$ ，保证 $1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq w_i \leq 10^9, 1 \leq c_1, c_2, \dots, c_n \leq 10^9$ 。

- Subtask 1 (20) : $n \leq 10$ 。
- Subtask 2 (30) : $m = n - 1$ 。
- Subtask 3 (20) : $n \leq 1000$ 。
- Subtask 4 (30) : 无特殊限制。

Problem C. 古墓丽影：崛起 (rise.c/cpp)

时间限制：1s

空间限制：512MB

“我这一生.....没见过几个像你这么出色的人。我的人生终于.....要结束了。”

“对不起.....我原本只是想为这个世界做点什么。”

“你已经做到了.....你已经做到了。”

“我只希望有一天你能以自己的方式留名于世。毕竟重要的不是我们的身份，而是我们做过的事情。”

小 K 最近在玩一款叫做《古墓丽影：崛起》(Rise of the Tomb Raider) 的游戏。在这款游戏中，小 K 需要操纵年轻的冒险家劳拉·克劳馥，穿越重重雪山，粉碎邪恶组织圣三一支配世界的阴谋。

现在，劳拉正在攀爬一系列连绵的雪山。在她面前有两条并列的山脉，山脉 A 和山脉 B，各自包含 $2n$ 座山峰，海拔分别为 a_1, a_2, \dots, a_{2n} 和 b_1, b_2, \dots, b_{2n} 。她装备了一套新型的机动装置，对于任意的 x ，她都可以从任意一条山脉的第 x 座山峰移动到任意一条山脉的第 $(x+1)$ 座山峰上去，但为了保持平衡，她必须保证途径的所有山峰海拔**不降**（非严格，即可以存在相等的海拔）。她可以选择在任意一条山脉的第 1 座山峰开始攀登，同时可以选择在任意一条山脉的第 $2n$ 座山峰结束攀登。可以看出，她会经过恰好 $2n$ 座山峰，其中一些来自山脉 A，另一些来自山脉 B。为了更全面地认识两条山脉的环境，探寻其中隐藏的秘密与宝藏，她希望经过恰好 n 座 A 山脉的山峰和 n 座 B 山脉的山峰。

劳拉想要知道，是否存在这样的方案。如果存在，她还希望你能帮她构造出**任意一组**满足条件的方案来。

Input

输入从文件 `rise.in` 中读取。

第一行包含一个数字 n ，表示山脉长度的一半。

第二行包含 $2n$ 个正整数 a_1, a_2, \dots, a_{2n} ，表示 A 山脉的所有山峰的海拔。

第三行包含 $2n$ 个正整数 b_1, b_2, \dots, b_{2n} ，表示 B 山脉的所有山峰的海拔。

Output

输出到文件 `rise.out` 中。

如果有解，直接输出一行长度为 $2n$ 的字符串 s 。如果劳拉经过的第 i 座山峰来自 A 山脉，则 $s_i = A$ ，否则 $s_i = B$ 。

如果无解，输出一行一个数 -1 。

Sample Input 1

```
3
2 5 4 9 15 11
6 7 6 8 12 14
```

Sample Output 1

```
AABABB
```

Sample Explanation 1

对应的海拔序列为 2, 5, 6, 9, 12, 14，满足单调不降的性质。

Sample Input 2

```
2
1 4 10 20
3 5 8 13
```

Sample Output 2

```
BBAA
```

Sample Explanation 2

不难发现，对于这组数据来说，任意一个包含恰好 2 座 A 山脉的山峰的方案都是可行的。你只需要输出其中的任意一种。

Sample Input 3

```
2
3 4 5 6
10 9 8 7
```

Sample Output 3

```
-1
```

Sample Input 4

```
6
25 18 40 37 29 95 41 53 39 69 61 90
14 18 22 28 18 30 32 32 63 58 71 78
```

Sample Output 4

```
BABBABAABABA
```

Subtasks & Constraints

本题采用子任务制评测，你能得到该子任务的分数当且仅当你通过了这个子任务中的**所有**测试点。对所有的数据，保证 $1 \leq n \leq 5 \times 10^5$ ，对任意的 $i \in [1, 2n] \cup \mathbb{Z}$ ，保证 $1 \leq a_i, b_i \leq 10^9$ 。

- Subtask 1 (20) : $n \leq 6$ 。
- Subtask 2 (30) : $n \leq 2000$ 。
- Subtask 3 (50) : 无特殊限制。

Problem D. 艾尔登法环 (ring.c/cpp)

时间限制：2s

空间限制：512MB

“老头环癔发作最严重的一次，躺在床上，拼命念赞美太阳，难受得一直打铁，以为打只狼没事，看到全苇名染上龙咳，抖刀越来越快都要活尸化了一样，满头是虚汗，身上感觉有征战骑士在爬，全地图每一个地方都在死.....我真的想玩老头环想得发疯了。我开卡丁车会想老头环，我喝原素瓶会想老头环，我忍杀会想老头环，我翻滚会想老头环，我枪反会想老头环，我坐篝火会想老头环，我佛渡会想老头环，我盯着 steam 的老头环看，我盯着万代南梦宫的老头环看，我盯着街机里的老头环看.....我每时每刻眼睛都直直地盯着老头环看，像一个蛇眼一样扫视经过我身边的每一个 Elden Ring，我真的觉得自己像磕了绿花草了一样，我对老头环的念想似乎都是病.....”

小 M 最近一直在等一款叫做《艾尔登法环》(Elden Ring) 的游戏发售。小 M 等的时间太长了，一直等到她曾经精心打理的花园变得荒草丛生，满目疮痍。她觉得自己不能再这样盲目地等下去了，于是决定先从重新打理自己的花园开始。

第 0 天清晨，小 M 买来了等量的 n 种植物种子，将它们播撒在了除去荒草的花园里。她知道，第 i 种植物每一天会长高 a_i 个单位（无论它们处在什么样的状态），而它们在第 0 天都是 0 单位高。她决定要经常打理花园，因此她制定了接下来一段时间里，对花园的 m 次修剪计划。第 i 次修剪会发生在第 d_i 天清晨，她会将所有植物高于 b_i 单位的部分全部剪掉。被剪下来的植物片段有各种用途，因此她希望将所有被剪掉的片段收集起来。注意，每次修剪计划都会被执行，因此在计算第 i 次修剪的结果时，你需要考虑第 1 至第 $(i - 1)$ 次修剪的影响。保证 $d_1 < d_2 < \dots < d_m$ ，也就是说小 M 会按顺序给出所有修剪计划，且同一天最多修剪一次。

现在，小 M 想知道，每次修剪计划中，被剪下来的片段总长度是多少个单位。

Input

输入从文件 `ring.in` 中读取。

第一行两个正整数 n, m ，分别表示植物种类数和修剪计划数。

第二行 n 个正整数 a_1, a_2, \dots, a_n ，表示 n 种植物各自的生长速率。

接下来 m 行，第 i 行两个正整数 d_i, b_i ，分别表示第 i 次修剪计划发生在第 d_i 天，这一天小 M 将会剪掉所有植物高于 b_i 的部分。

Output

输出到文件 `ring.out` 中。

输出 m 行，第 i 行一个正整数，表示第 i 天小 M 剪下来的植物片段的总长度。

Sample Input 1

```
4 4
1 2 4 3
1 1
2 2
3 0
4 4
```

Sample Output 1

```
6
6
18
0
```

Sample Explanation 1

天数	修剪前高度	修剪后高度
1	1, 2, 4, 3	1, 1, 1, 1
2	2, 3, 5, 4	2, 2, 2, 2
3	3, 4, 6, 5	0, 0, 0, 0
4	1, 2, 4, 3	1, 2, 4, 3

Sample Input/Output 2

见选手下发文件中 `ex_ring2.in/ex_ring2.ans`。

Subtasks & Constraints

本题采用子任务制评测，你能得到该子任务的分数当且仅当你通过了这个子任务中的**所有**测试点。对所有的数据，保证 $1 \leq n, m \leq 5 \times 10^5$, $1 \leq a_1, a_2, \dots, a_n \leq 10^6$, $1 \leq d_1 < d_2 < \dots < d_m \leq 10^{12}$, $0 \leq b_1, b_2, \dots, b_m \leq 10^{12}$ 。保证任意时刻不存在高度高于 10^{12} 的植物。

- Subtask 1 (20) : $n, m \leq 5000$ 。
- Subtask 2 (20) : $b_i \leq 100$ 。
- Subtask 3 (30) : $n, m \leq 10^5$ 。

- Subtask 4 (30) : 无特殊限制。