

A 聚会

时空限制：1s 256MB

文件名

party.in/party.out/party.cpp

题意

有一些人需要聚集在一起聚会, 每个人初始都位于 n 个排列在一条直线上的某个点 p_i 上, 第 i 个点上有 a_i 个人。你需要从这些 p_i 中找出一个点 p_k , 使得所有人到达 p_k 移动的距离之和最小。

输入格式

第一行输入一个整数 n ($1 \leq n \leq 10^6$), 表示点的个数 接下来 n 行, 每行两个整数 p_i, a_i ($1 \leq p_i, a_i \leq 10^6, p_1 < p_2 < p_3 < \dots < p_n$), 分别表示点的位置和位于该点上的人数。

输出格式

输出一行一个整数, 表示所有人移动距离之和的最小值。

输入样例

```
4
1 3
3 1
7 2
8 4
```

输出样例

```
26
```

样例解释

选择 $k = 3$, 移动的距离之和为 $(7 - 1) \times 3 + (7 - 3) \times 1 + (7 - 7) \times 2 + (8 - 7) \times 4 = 26$

B 青蛙

时空限制：1s 256MB

文件名

frogs.in/frogs.out/frogs.cpp

题意

有 m 颗石头排成一个环, 编号为 0 到 $m - 1$ 。有 n 只青蛙在 0 号石头上, 第 i 只青蛙每次能恰好跳过 a_i 颗石头。具体地, 如果第 i 只青蛙当前在编号 x 的石头上, 它下一跳会到达 $(x + a_i) \bmod m$ 的石头 (因为石头是成环排列的)。

青蛙 i 每跳到一颗石头上, 这颗石头都会被青蛙 i 占领, 即使这只青蛙跳出了这颗石头, 石头还是被青蛙 i 占领。青蛙们现在想知道, 当它们跳跃次数足够多之后, 最终有多少颗石头被两只及以上的青蛙占领。因为这样的石头个数可能很多, 你只需要输出这些石头的编号之和。

输入格式

第一行输入两个整数 n, m ($1 \leq n \leq 10^4, 1 \leq m \leq 10^9$), 分别表示青蛙数和石头数。

第二行输入 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), 表示每只青蛙每次跳过的石头数。

输出格式

输出一行一个整数, 表示至少会被两只青蛙占领的石头编号之和。

输入样例1

```
2 12
9 10
```

输出样例1

```
42
```

输入样例2

```
9 96
81 40 48 32 64 16 96 42 72
```

输出样例2

1872

C 分层图最短路

时空限制：1s 256MB

文件名

shortest.in/shortest.out/shortest.cpp

题意

给一张 n 个点的无向图, 其中每一个点都在某一层 l_i 内, 对于不同的层, 我们还可以花费 C 的代价从层 x 的某一个点到达层 $x + 1$ 的某一个点 (如果层 $x + 1$ 没有点, 则不能进行此操作), 当然也可以从层 $x + 1$ 的某一点花费 C 的代价到达层 x 的某一点 (如果层 x 没有点, 则不能进行此操作)。此外, 还有 m 条特殊的双向边, 每条边连接两个不同的点 u_i, v_i , 走这条边的花费是 w_i 。你需要求出从1号点到达 n 号点的最小花费。注意: 如果点 u, v 在同一层, 且 u, v 之间没有连边, 那么你不能从 u 花费 0 的代价直接到 v 。

输入格式

第一行输入三个整数 n, m, C ($1 \leq n, m \leq 10^5, 1 \leq C \leq 10^3$), 分别表示点数、边数和在两个相邻层之间的移动花费。接下来一行输入 n 个整数 l_1, l_2, \dots, l_n ($1 \leq l_i \leq n$), 分别表示每个点所在的层。

接下来 m 行, 每行输入三个整数 u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^4$), 描述额外的 m 条双向边。

输出格式

输出一行一个整数, 表示从节点 1 到节点 n 的最小花费。如果你不能从 1 到达 n , 则输出 -1 。

输入样例1

```
4 3 3
1 2 2 3
1 4 1000
1 2 10
1 3 2
```

输出样例1

5

输入样例2

3 1 3
1 3 3
1 2 4

输出样例2

-1