

PION 模拟赛

题目名称	数字	连接	关键点	披萨
题目类型	传统型	传统型	传统型	传统型
英文名	num	connect	key	pizza
源程序名	num.cpp	connect.cpp	key.cpp	pizza.cpp
输入文件名	num.in	connect.in	key.in	pizza.in
输出文件名	num.out	connect.out	key.out	pizza.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	0.5 秒
内存限制	1 GB	1 GB	1 GB	1 GB
测试点数量	14 个	20 个	子任务	子任务
编译选项	-lm -O2 -std=c++14 -Wl,--stack=536870912			

【注意事项（请仔细阅读）】

1. 选手提交的源程序必须存放在以**选手编号命名**的文件夹中，文件名称与对应试题英文名一致，**不要创建子文件夹**。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
4. 若无特殊说明，结果比较方式为**忽略行末空格、文本回车后的全文比较**。
5. 程序中使用的栈空间大小同样受到内存限制。
6. 每道题目提交的**代码文件大小限制为 100KB**。

1 数字 (num)

1.0s / 1.0GB / 14 个测试点

给定正整数 X, A, B ，以及集合 $S \subset \{0, 1, \dots, 9\}$ ，问有多少个正整数 y 满足以下三个条件：

- $y \in [A, B]$ ；
- y 是 X 的倍数；
- 十进制下， y 的每一位数字都属于集合 S 。

输入格式

输入第一行包含三个整数 X, A, B 。

第二行为一个数字串表示集合 S ，保证数字不会重复出现。

输出格式

输出一个数表示答案。

样例 #1

Input	Output
2 1 20 0123456789	10

样例 #2

Input	Output
6 100 9294 23689	111

样例 #3

Input	Output
5 4395 9999999999 12346789	0

数据范围与约定

对于 100% 的数据，保证 $1 \leq X < 10^{11}$ ， $1 \leq A \leq B < 10^{11}$ 。

2 连接 (connect)

1.0s / 1.0GB / 20 个测试点

题目描述

平面上有 n 个点，第 i 个点坐标为 (x_i, y_i) ，权值为 k_i 。

现在，你需要指定一个实数 D 。当你指定 D 后，所有欧式距离不超过 D 的两个点之间会连边。如果连边之后，存在一些点的权值之和是 K 的倍数，且它们位于同一连通块内，那么 E 老板就会感到开心。

请确定最小的 D 使 E 老板感到开心。

输入格式

第一行两个整数 n, K

接下来 n 行，每行三个整数 x_i, y_i, k_i ，含义如题。

输出格式

输出一个实数表示答案，四舍五入保留三位小数。

样例 #1

Input		
3	3	
0	4	4
1	5	1
2	6	1

Output		
1.414		

样例 #2

Input		
6	11	
0	0	1
0	1	2
1	0	3
1	1	4
5	5	1
20	20	10

Output		
5.657		

数据范围与约定

对于 40% 的数据，有 $1 \leq n \leq 10^3$

对于 100% 的数据，有 $1 \leq n \leq 5 \times 10^4$, $1 \leq K \leq 30$, $0 \leq x_i, y_i, k_i \leq 10^8$ ，且 k_i 不是 K 的倍数。

输入数据保证有解。

3 关键点 (key)

1.0s / 1.0GB / 子任务

题目描述

给定一棵 n 个点的基环树，你需要指定其中一些点为关键点，使得对任意一个点 u ，与它相邻的点集 $\mathcal{N}(u)$ 中有且仅有一个关键点。

为了达成目标，你至少需要指定几个关键点？如果无法达成目标，输出 -1 。

输入格式

第一行一个正整数 n 表示点数。

接下来 n 行，每行两个整数 u, v 表示一条边。

输出格式

输出一个数表示答案。如果无法达成目标，输出 -1 。

样例 #1

Input
4
1 2
2 3
3 4
4 1

Output
2

样例 #2

Input
3
1 2
2 3
3 1

Output
-1

更多样例见下发文件。

数据范围与约定

本题采用子任务捆绑测试

Subtask 1 (10 分): 每个点的度数都为 2

Subtask 2 (10 分): $n \leq 20$

Subtask 3 (40 分): $n \leq 10^3$

Subtask 4 (40 分): 无特殊限制

对于所有数据，保证 $3 \leq n \leq 10^5$

4 披萨 (pizza)

0.5s / 1.0GB / 子任务

题目描述

E 老板开了一家披萨店。但 E 老板太穷了，所以他只买了一个烤箱，因此无法同时制作多于一个披萨。

一天的时刻从 0 开始。现在有 N 个订单，第 i 个订单的顾客要求在 L_i 时刻来店就餐，而他所需的披萨需要花费 T_i 时间单位来制作。

如果一个顾客的披萨提前了 K 个时间单位做好，那么顾客会感到开心，E 老板可以获得 K 元小费；反之，如果一个顾客的披萨延迟了 K 个时间单位才做好，根据延时宝条例，E 老板需要向顾客赔付 K 元。

假设 E 老板一天共赚 A 元小费，通过延时宝一共向顾客赔付 B 元。请你帮助 E 老板安排制作披萨的顺序，使得 $A - B$ 最大。

然而，顾客有时候因为个人原因需要修改订单，一次只会有一个顾客修改。请你帮助 E 老板求出每次修改订单后 $A - B$ 的最大值。

输入格式

第一行两个整数 N, C ，表示订单数和修改订单的次数。

接下来 N 行，每行两个整数 L_i, T_i ，含义如题。

接下来 C 行，每行三个正整数 i, L_i, T_i ，表示 i 号顾客要修改订单，以及修改后的数字。

输出格式

输出共 $C + 1$ 行。一行一个整数表示 $A - B$ 的最大值。

在未修改订单之前，你需要求 $A - B$ 的最大值；接下来每次修改后，你需要再求一次 $A - B$ 的最大值。

样例 #1

Input	Output
3 2 10 2 6 5 4 3 1 6 1 3 0 10	3 2 -11

更多样例见下发文件。

数据范围与约定

Subtask 1 (50 分): $1 \leq T_i \leq 10^3$

Subtask 2 (50 分): 无特殊限制。

对于 100% 的数据， $1 \leq N, C \leq 2 \times 10^5$ ， $0 \leq L_i, T_i \leq 10^5$ 。