

图灵编程 10 月 20 日普及组模拟赛

图灵编程 10 月 20 日模拟赛

时间：2024 年 10 月 20 日 18:10 ~ 21:30

题目名称	重新覆写	电子天使	如果能成为谁的 心脏的话	机械之声
题目类型	传统型	传统型	传统型	传统型
可执行文件名	teto.exe	ral.exe	gumi.exe	voice.exe
输入文件名	teto.in	ral.in	gumi.in	voice.in
输出文件名	teto.out	ral.out	gumi.out	voice.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	20	20	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	teto.cpp	ral.cpp	gumi.cpp	voice.cpp
-----------	----------	---------	----------	-----------

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
-----------	--------------------------------------

重新覆写 (teto)

【题目背景】

要不，来点不一样的内容？此处，没有“我”，只有一些不存在于现实的“生命”，因而，你需要做的事情也会与之前有所偏差。

当然，我们还是在比赛，还是在做题，区别很大吗？倒也不至于。

【题目描述】

teto 有 $n(n \leq 10^5)$ 天的休息天，一开始这几天都是在休假。她想要在这 n 天中完成 Q 件任务。

对于第 i 个任务来说，teto 计划在 $a_i \sim b_i$ 的这几天中做事情 c_i 。当然，可能会出现同一天需要干很多事情的情况，teto 决定只干计划中最后一件事情。

但是，毕竟是休息天，她还是想多休息的，于是全部计划安排完毕后，选择一个区间 $[x, y]$ ，在这个区间中接下来无论这几天原本干的是什么事情，统统不干了，改成休假。

那么，她在最后每一天分别做了什么事情呢？

【输入格式】

从文件 *teto.in* 中读入数据。

第一行包含两个正整数， n 和 Q 。

第二行到第 $n+1$ 行，每行有三个正整数，分别是 a, b, c

接下来一行包含两个正整数， x 和 y

【输出格式】

输出到文件 *teto.out* 中。

一共一行，其中第 i 个数为 tetoto 第 i 天做的事情，如果 tetoto 这天没做事情，输出 0。

【样例 1 输入】

```
1 6 3
2 1 2 3
3 4 5 6
4 2 4 4
5 3 4
```

【样例 1 输出】

```
1 3 4 0 0 6 0
```

【样例 1 解释】

对于 teto 的计划而言：

- 第一天做事情 3。
- 第二天做事情 3, 4, 实际只做最后一件事情, 做事情 4。
- 第三天做事情 4, 最后选择休息。
- 第四天也选择休息。
- 第五天做事情 6。
- 第六天没有考虑做任何事情, 在休息。

【样例 2】

见选手目录下的 *teto/teto2.in* 与 *teto/teto2.ans*。

【子任务】

令 $m = n - (y - x + 1)$ 。

对于 35% 的数据, $n, Q \leq 1000$

另有 15% 的数据, $m = 0$

另有 20% 的数据, $m = 1$

对于 100% 的数据, $n, Q \leq 10^5; a \leq b \leq n; x \leq y \leq n; m \leq 100; c \leq 10^6$

电子天使 (ral)

【题目背景】

你若是问现实与虚拟的区别在哪里？差距很大，不是吗？但是如今时代，你能不离开虚拟吗？我们的 OJ 是虚拟的，荧幕中映射出来的，也是虚拟的，现代社会，无法脱离，但它，永不真实。

可能会有人，会喜欢上这样的时代吧？但无论是否喜欢，是否厌弃，时代将不断前移，只能适应，无法逃离。

【题目描述】

C 市的节庆将近，小天使 len 和 rin 准备给 C 市的每一个人分发礼物。

C 市有 $n(n \leq 10^5)$ 栋建筑，分布在 $[1 - 10^9]$ 的数轴上，对于第 i 栋建筑来说，其位于 $x_i(x_i$ 不会重复) 的位置上，要为里面的人分发礼物需要 t_i 的时间，因为如果两个人同时进入一栋楼，可能会重复给一个人送礼，因此他们不会走到同一栋建筑中

rin 和 len 在一开始都可以任意选择一个位置，然后直接传送到对应的位置，接下来就只能通过步行一步步走过去了，他们每一秒都可以走一格（也就是从 x 走到 $x + 1$ 或者 $x - 1$ ）。

在这样的情况下，rin 和 len 想知道，要给所有人都发完礼物，至少需要多少时间？

【输入格式】

从文件 *ral.in* 中读入数据。

第一行有一个正整数，为 n

第二行有 n 个正整数，第 i 个正整数为 x_i

第三行有 n 个正整数，第 i 个正整数为 t_i

【输出格式】

输出到文件 *ral.out* 中。

输出一个正整数，为发完礼物最少需要的时间。

【样例 1 输入】

```
1 4
2 7 3 5 1
3 2 3 2 2
```

【样例 1 输出】

```
1 7
```

【样例 1 解释】

一种可行的方法：

rin 传送到 4，然后走到 5，为第三个建筑里面的人发礼物，然后走到 7，为第一个建筑里面的人发礼物。

rin 走路时间为 $1 + 2 = 3$ ，送礼物时间为 $2 + 2 = 4$ ，花费时间为 $3 + 4 = 7$ 。

len 传送到 3，然后为第二个建筑里面的人发礼物，然后走到 1，为第四个建筑里面的人发礼物。

len 走路时间为 2，送礼物时间为 $3 + 2 = 5$ ，花费时间为 $2 + 5 = 7$

花费的时间为 $\max(7, 7) = 7$ 。

方法不唯一，但是可以证明这种方法是一种最优解

【样例 2 输入】

```
1 6
2 10 5 7 8 1 120
3 50 20 10 22 3 6
```

【样例 2 输出】

```
1 114
```

【样例 3】

见选手目录下的 *ral/ral3.in* 与 *ral/ral3.ans*。

【子任务】

数据点	n	x_i	t_i
1-3	$n \leq 10$	$x_i \leq 150$	$t_i \leq 150$
4	$n \leq 10$	$x_i \leq n$	$t_i \leq 150$
5	$n \leq 10$	$x_i \leq 150$	$t_i = 1$
6-8	$n \leq 1000$	$x_i \leq 10^4$	$t_i \leq 10^4$
9	$n \leq 1000$	$x_i \leq n$	$t_i \leq 10^9$
10	$n \leq 1000$	$x_i \leq 10^9$	$t_i = 1$
11-13	$n \leq 10^5$	$x_i \leq n$	$t_i = 1$
14-15	$n \leq 10^5$	$x_i \leq n$	$t_i \leq 10^9$
16-17	$n \leq 10^5$	$x_i \leq 10^9$	$t_i = 1$
18-20	$n \leq 10^5$	$x_i \leq 10^9$	$t_i \leq 10^9$

对于 100% 的数据, $n \leq 10^5, x_i, t_i \leq 10^9$

如果能成为谁的心的话 (gumi)

【题目背景】

但是你要说人们之间的情感会因为隔着虚拟、网络的这一层隔阂，而丧失沟通吗？倒也未必，至少我们仍然会报以期许，愿在其中得到美好。

我们并不因为网络而失去任何事物，只是我们本身就太过渺小，在欢欣于万物之主的胜利的阴影中，存在着比任何一个人都要庞大的巨物。我们只是茫然于这样的利维坦，在对比下，感到了自己的一无所有。

【题目描述】

GUMI 想要帮助他人，但是她需要先有一颗自己的“心”，于是你需要帮助她组成“心”。

“心”分为 4 个部分，每个部分的初始值都是 0，你需要按照流程将“心”的任意一个部分的值大于等于 n ：

- 一：让 [当前值最高的所有]* 部分都 $\times x$ （这里的所有运算来说， $a + b$ 都相当于 $a + = b$ ）。
- 二：让第 1 个部分的值 $+y$ ，第 3 个部分的值 $-$ 第 2 部分的值（除法全部向零取整，如果除 0 不计算）。
- 变体流程一
- 三：找到 [第一个最大的]** 部分，[第一个最小的]*** 部分（重复只算一次），给这些部分赋值为这些部分向零取整的平均值。
- 变体流程二
- 四：让 [所有]**** 部分 $+4$
- 变体流程三
- 变体流程四

当一遍流程结束后，若没有达到要求，则重复这个流程，在流程中的任意一步结束后如果“心”的任意一部分的值大于等于 n ，则直接结束这个流程。

变体流程只有当对应的控制数字为 1 时才启用，具体内容如下：

- 变体流程一：顺时针旋转全部的运算符 45 度，也就是说 $+$ 变成 \times ， \times 变成 $+$ ， $-$ 变成 \backslash ($a \backslash b$ 等效于 b/a ，也即是说 $a \backslash b$ 相当于 $b / = a$)， \backslash 变成 $|$ (位运算的或)，依次类推
- 变体流程二：对于流程中对于部分的描述（当前最高的所有，第一个最大的，第一个最小的，所有，前文已经用中括号标记）移动到上一个位置，第一个描述移动到最后一个位置
- 变体流程三：所有流程中的阿拉伯数字（0-9 这种，不是一到九这种数字）移动到规则的下一个位置，最后一个阿拉伯数字移动到第一个位置
- 变体流程四：如果此时有一个部分小于等于 $-n$ ，输出 T_T，然后结束程序

为方便你的代码实现，你可以只实现部分要求，以取得部分分。

【输入格式】

从文件 *gumi.in* 中读入数据。

第一行包含一个正整数 n

第二行包含四个只会是 0 和 1 的数，第 i 个数为 0 时，第 i 条变体规则不会启动，否则，启动这条变体流程

第三行包含两个非零整数，为 x 和 y 。

【输出格式】

输出到文件 *gumi.out* 中。

依次输出结束时四个部分对应的数字，如果为满足变体流程四，改为输出 T_T

【样例 1 输入】

```
1 10
2 0 0 0 0
3 2 3
```

【样例 1 输出】

```
1 10 10 4 4
```

【样例 1 解释】

按照流程，其会依次变成

第 1 步：让当前最高的所有部分 $\times 2$

$\{0 \times 2, 0 \times 2, 0 \times 2, 0 \times 2\} \Rightarrow \{0, 0, 0, 0\}$

第 2 步：让第 1 个部分 $+3$ ，第三部分 $-$ 第二部分

$\{0 + 3, 0, 0 - 0, 0\} \Rightarrow \{3, 0, 0, 0\}$

第 3 步：找到第一个最大的部分（第一个部分），第一个最小的部分（第二个部分），给这些部分赋予部分的平均值，也就是 $(3 + 0)/2 = 1$

$\{3, 3, 0, 0\} \Rightarrow \{1, 1, 0, 0\}$

第 4 步：所有部分 $+4$

$\{1 + 4, 1 + 4, 0 + 4, 0 + 4\} \Rightarrow \{5, 5, 4, 4\}$

第 5 步：最大的所有部分 $\times 2$

$\{5 \times 2, 5 \times 2, 4, 4\} \Rightarrow \{10, 10, 4, 4\}$

【样例 2 输入】

```

1 10
2 1 1 1 1
3 2 3

```

【样例 2 输出】

```

1 12 4 4 4

```

【样例 2 解释】

请仔细阅读每一步具体执行的步骤，避免理解错误

按照流程，第 1 步为 $\{0 \times 2, 0 \times 2, 0 \times 2, 0 \times 2\} = \{0, 0, 0, 0\}$

第 2 步让第一个部分 +3，第三个部分减去第二个部分的值，变为 $\{0 + 3, 0, 0 - 0, 0\} = \{3, 0, 0, 0\}$

第 3 步旋转运算符。

第 4 步最大值为 3，最小值为 0，平均值为 1，四个部分变成 $\{1, 1, 0, 0\}$

第 5 步移动部分描述

第 6 步当前值最高的所有部分为前两个数，将其 $\times 4$ ，变成 $\{4, 4, 0, 0\}$

第 7 步移动阿拉伯数字

现在重新展示经过变体流程转换后的流程

- 一：让 [第一个最大的]* 部分都 $+x$ 。
- 二：让第 4 个部分的值 $\times y$ ，第 1 个部分的值 3 部分的值（除法全部向零取整，如果除 0 不计算）。
- 三：找到 [第一个最小的]** 部分，[所有]** 部分（重复只算一次），给这些部分赋值为这些部分向零取整的平均值。
- 四：让 [当前值最高的所有]** 部分 $\times 2$

第 8 步执行变体流程四

第 9 步让第一个最大部分的 +2，变成 $\{6, 4, 0, 0\}$

第 10 步让第 4 个部分 $\times 3$ ，第 1 个部分 \ 第 3 个部分，变成 $\{6, 4, 0/6, 0 \times 3\} = \{6, 4, 0, 0\}$

第 11 步旋转运算符。

第 12 步找到所有部分和第一个最小的部分，对这些取平均值，和为 10，平均值为 2，变成 $\{2, 2, 2, 2\}$

第 13 步移动部分描述。

第 14 步让第一个最大的数 +2，变成 $\{4, 2, 2, 2\}$

第 15 步移动阿拉伯数字

现在重新展示经过变体流程后的流程

- 一：让 [第一个最小的]* 部分都 $\times x$ 。
- 二：让第 2 个部分的值 $+y$ ，第 4 个部分的值 | 第 1 部分的值（除法全部向零取整，如果除 0 不计算）。
- 三：找到 [所有]** 部分，[当前值最高的所有]** 部分（重复只算一次），给这些部分赋值为这些部分向零取整的平均值。
- 四：让 [第一个最大的]** 部分 $+3$

第 16 步让第一个最小的部分 $\times 2$ ， $\{4, 2 \times 2, 2, 2\} \Rightarrow \{4, 4, 2, 2\}$

第 17 步让第 2 个部分 $+3$ ，第 4 个部分的值 | 第 1 部分的值， $\{4, 4+3, 2, 2|4\} \Rightarrow \{4, 7, 2, 6\}$

第 18 步旋转运算符

第 19 步找到所有部分，当前值最高的所有部分，对这些取平均值， $(4+7+2+6)/4 = 4$ ，四个部分变成 $\{4, 4, 4, 4\}$

第 20 步移动部分描述

第 21 步让第一个最小的部分 $\times 3$ ， $\{4 \times 3, 4, 4, 4\} \Rightarrow \{12, 4, 4, 4\}$ ，此时第一个部分大于 10，流程结束。

展示此时流程

- 一：让 [所有]* 部分都 $+x$ 。
- 二：让第 2 个部分的值 $\times y$ ，第 4 个部分的值 / 第 1 部分的值（除法全部向零取整，如果除 0 不计算）。
- 三：找到 [当前值最高的所有]** 部分，[第一个最大的]** 部分（重复只算一次），给这些部分赋值为这些部分向零取整的平均值。
- 四：让 [第一个最小的]** 部分 $\times 3$

需要注意对于流程四来说，由于前面经过了两个变体流程，与展示的流程是不完全一样的。

【样例 3 输入】

```
1 6116
2 0 0 0 1
3 -42 -1
```

【样例 3 输出】

```
1 T_T
```

【样例 3 解释】

程序结束时，四个部分分别是

-69777 -69777 4 -156

【样例 4】

见选手目录下的 *gumi/gumi4.in* 与 *gumi/gumi4.ans*。

【子任务】

数据点	变体流程一	变体流程二	变体流程三	变体流程四
1 - 4	0	0	0	0
5 - 6	1	0	0	0
7 - 8	0	1	0	0
9 - 10	0	0	1	0
11 - 12	1	1	0	0
13 - 14	0	1	1	0
15 - 16	1	0	1	0
17 - 18	0	0	0	1
19 - 20	1	1	1	1

对于 100% 的数据， $|x|, |y| \leq 100, n \leq 10^4$ ，给定数据不会出现死循环

机械之声 (voice)

【题目背景】

最终，在最关键时刻，还是以沉默做结局更好。

所谓的背景，不过是无所谓的空谈，题面中那些多余的内容，亦不过是一些提升你读题能力无聊的手段。

题目有必要写的精彩吗？我只能回答到：我不知道。

说这是题目，不是作文，只需要让人读懂即可，除此以外，都是累赘。

但是这些多余的内容，可能还是会有点意思吧！

随意的话语结束，该到最后了。

【题目描述】

音乐的同位体的 5 名歌手们要共同为一首歌歌唱，对于这 5 名歌手来说，其自身的嗓音存在一个音率度，第 i 名歌手拥有 B_i 的音律度。

这首歌一共有 n 个部分，每个部分都需要安排一名歌手。

对于每个部分来说，由于歌词以及旋律的分配，其存在一个 A 的音节度。

对于第 i 名歌手来说，如果其歌唱了第 j 部分的歌曲，这部分歌曲的音节度将会变成 $A_j + B_i$ 。

对于一首歌曲来说，应该要做到尽可能的均衡，不要出现相邻的部分音节度差值太大的情况。

因此，音乐的同位体的成员们想要知道，如何分配每个人歌唱的部分，让相邻的部分的音节度所有的差值都尽可能小？

当然，在任何一种要求下，人员分配的结果是会不同的，你需要找到的是所有相邻部分的音节度的差值和最小的分配方法。

【输入格式】

从文件 *voice.in* 中读入数据。

第一行输入一个正整数 n

第二行输入 5 个正整数，分别是这 5 名歌手的音律度

第三行一个 n 个正整数，第 i 个数字为第 i 部分的音节度 A_i

【输出格式】

输出到文件 *voice.out* 中。

本题采用 spj，两行正确任意一行都可以拿到一半的分数，但是如果想要拿到一半的分数，也需要输出全部对应的内容（即使是错误的）

第一行输出所有相邻部分的音节度的差值的和。

第二行输出 n 个数，第 i 个数为第 i 部分安排的歌手，任意一种最优解都视为正确解。

【样例 1 输入】

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

【样例 1 输出】

```
1 0
2 2 5 3
```

【样例 1 解释】

最后这首歌的音节度变为 7,7,7，差值为 0。

方法不唯一，但是可以证明这是最优解。

【样例 2】

见选手目录下的 *voice/voice2.in* 与 *voice/voice2.ans*。

【子任务】

对于 30% 数据， $n \leq 10$

另有 20% 的数据，所有的 A 满足单调性

另有 20% 的数据， $B_1 = B_2, B_3 = B_4 = B_5$

对于 100% 的数据， $n \leq 10^5; A_i, B_i \leq 10^7$