第 33 届全国信息学奥林匹克竞赛

CCF NOI 2016

第二试

竞赛时间：2016 年 7 月 26 日 8:30-13:30

题目名称	区间	国王饮水记	旷野大计算
目录	interval	drink	nodes
可执行文件名	interval	drink	N/A
输入文件名	interval.in	drink.in	nodes1.in~ nodes10.in
输出文件名	interval.out	drink.out	nodes1.out~ nodes10.out
每个测试点时限	3 秒	1.5 秒	N/A
内存限制	256MB	256MB	N/A
测试点数目	20	20	10
每个测试点分值	5	5	10
是否有部分分	否	是	是
题目类型	传统型	传统型	提交答案型
是否有样例文件	是	是	否
是否有附加文件	否	否	是

提交源程序须加后缀

对于 C++　语言	interval.cpp	drink.cpp	N/A
对于 C　　语言	interval.c	drink.c	N/A
对于 Pascal 语言	interval.pas	drink.pas	N/A

编译开关

对于 C++　语言	-O2 -lm	-O2 -lm	N/A
对于 C　　语言	-O2 -lm	-O2 -lm	N/A
对于 Pascal 语言	-O2	-O2	N/A

区间

【问题背景】

在数轴上有 n 个闭区间 $[l_1, r_1], [l_2, r_2], \ldots, [l_n, r_n]$。现在要从中选出 m 个区间，使得这 m 个区间共同包含至少一个位置。换句话说，就是使得存在一个 x，使得对于每一个被选中的区间 $[l_i, r_i]$，都有 $l_i \le x \le r_i$。

对于一个合法的选取方案，它的花费为**被选中的最长区间长度减去被选中的最短区间长度**。区间 $[l_i, r_i]$ 的长度定义为 $r_i - l_i$，即等于它的右端点的值减去左端点的值。

求所有合法方案中最小的花费。如果不存在合法的方案，输出 -1。

【输入格式】

输入文件为 ***interval.in***。

输入文件的第一行包含两个正整数 n, m，用空格隔开，意义如上文所述。保证 $1 \le m \le n$。

接下来 n 行，每行表示一个区间，包含用空格隔开的两个整数 l_i 和 r_i，为该区间的左右端点。

【输出格式】

输出文件为 ***interval.out***。

输出文件只有一行，包含一个正整数，即最小花费。

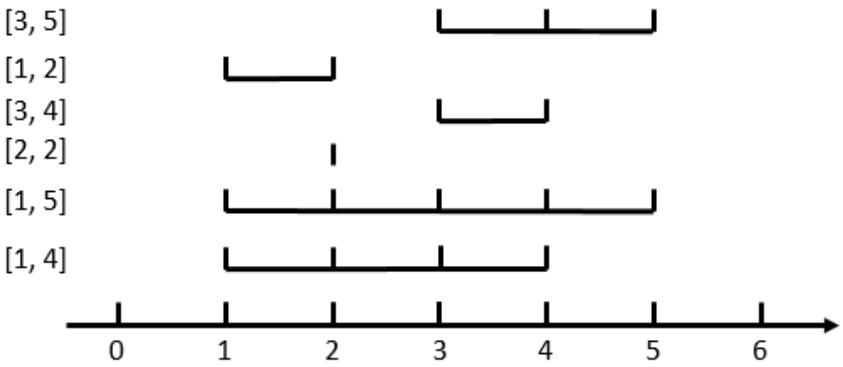
【样例 1 输入】

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

【样例 1 输出】

```
2
```

【样例 1 说明】



如图，当 $n = 6,\ m = 3$ 时，花费最小的方案是选取 [3,5]、[3,4]、[1,4] 这三个区间，他们共同包含了 4 这个位置，所以是合法的。其中最长的区间是 [1,4]，最短的区间是 [3,4]，所以它的花费是 $(4-1)-(4-3)=2$。

【样例 2 输入输出】

见选手目录下的 ***interval/interval2.in*** 与 ***interval/interval2.ans***。

【样例 3 输入输出】

见选手目录下的 ***interval/interval3.in*** 与 ***interval/interval3.ans***。

【子任务】

所有测试数据的范围和特点如下表所示：

测试点编号	n	m	l_i, r_i
1	20	9	$0 \le l_i \le r_i \le 100$
2	20	10	$0 \le l_i \le r_i \le 100$
3	199	3	$0 \le l_i \le r_i \le 100000$
4	200	3	$0 \le l_i \le r_i \le 100000$
5	1000	2	$0 \le l_i \le r_i \le 100000$
6	2000	2	$0 \le l_i \le r_i \le 100000$
7	199	60	$0 \le l_i \le r_i \le 5000$
8	200	50	$0 \le l_i \le r_i \le 5000$
9	200	50	$0 \le l_i \le r_i \le 10^9$
10	1999	500	$0 \le l_i \le r_i \le 5000$
11	2000	400	$0 \le l_i \le r_i \le 5000$
12	2000	500	$0 \le l_i \le r_i \le 10^9$
13	30000	2000	$0 \le l_i \le r_i \le 100000$
14	40000	1000	$0 \le l_i \le r_i \le 100000$
15	50000	15000	$0 \le l_i \le r_i \le 100000$
16	100000	20000	$0 \le l_i \le r_i \le 100000$
17	200000	20000	$0 \le l_i \le r_i \le 10^9$
18	300000	50000	$0 \le l_i \le r_i \le 10^9$
19	400000	90000	$0 \le l_i \le r_i \le 10^9$
20	500000	200000	$0 \le l_i \le r_i \le 10^9$

国王饮水记

【问题描述】

跳蚤国有 n 个城市，伟大的跳蚤国王居住在跳蚤国首都中，即 1 号城市中。

跳蚤国最大的问题就是饮水问题，由于首都中居住的跳蚤实在太多，跳蚤国王又体恤地将分配给他的水也给跳蚤国居民饮用，这导致跳蚤国王也经常喝不上水。

于是，跳蚤国在**每个**城市都修建了一个圆柱形水箱，这些水箱**完全相同**且**足够高**。一个雨天后，第 i 个城市收集到了高度为 h_i 的水。由于地理和天气因素的影响，**任何**两个不同城市收集到的水高度**互不相同**。

跳蚤国王也请来蚂蚁工匠帮忙，建立了一个庞大的地下连通系统。跳蚤国王每次使用地下连通系统时，可以指定**任意多**的城市，将这些城市的水箱用地下连通系统连接起来足够长的时间之后，再将地下连通系统**关闭**。由连通器原理，这些城市的水箱中的水在这次操作后会到达**同一高度**，并且这一高度等于指定的各水箱高度的**平均值**。

由于地下连通系统的复杂性，跳蚤国王**至多**只能使用 k 次地下连通系统。跳蚤国王请你告诉他，首都 1 号城市水箱中的水位最高能有多高？

【输入格式】

输入文件为 ***drink.in***。

输入的第一行包含 3 个正整数 n,k,p，分别表示跳蚤国中城市的数量，跳蚤国王能使用地下连通系统的最多次数，以及你输出的答案要求的精度。p 的含义将在输出格式中解释。

接下来一行包含 n 个正整数，描述城市的水箱在雨后的水位。其中第 i 个正整数 h_i 表示第 i 个城市的水箱的水位。保证 h_i 互不相同，$1 \le h_i \le 10^5$。

【输出格式与部分分】

输出文件为 ***drink.out***。

输出仅一行一个实数，表示 1 号城市的水箱中的最高水位。

这个实数只可以包含非负整数部分、小数点和小数部分。其中非负整数部分为必需部分，不加正负号。若有小数部分，则非负整数部分与小数部分之间以一个小数点隔开。若无小数部分，则不加小数点。

你输出的实数在小数点后不能超过 $2p$ 位，建议保留至少 p 位。

数据保证参考答案与真实答案的绝对误差小于 10^{-2p}。

你的输出被判定为正确当且仅当你的输出与参考答案的绝对误差**小于** 10^{-p}。

此外，每个测试点你将**有可能获得部分分**。

如果你的输出与参考答案的绝对误差不小于 10^{-p} 但**小于** 10^{-5}，你可以获得该测试点40%的分数。

【样例 1 输入】

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

【样例 1 输出】

```
2.666667
```

【样例 1 说明】

由于至多使用一次地下连通系统，有以下 5 种方案：

1. 不使用地下连通系统：此时 1 号城市的水箱水位为 1。
2. 使用一次连通系统，连通 1、2 号：此时 1 号城市的水箱水位为 $\frac{5}{2}$。
3. 使用一次连通系统，连通 1、3 号：此时 1 号城市的水箱水位为 2。
4. 使用一次连通系统，连通 2、3 号：此时 1 号城市的水箱水位为 1。
5. 使用一次连通系统，连通 1、2、3 号：此时 1 号城市的水箱水位为 $\frac{8}{3}$。

【样例 2 输入】

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

【样例 2 输出】

```
3.000000
```

【样例 2 说明】

此时最优方案为使用两次连通系统，第一次连通 1、3 号，第二次连通 1、2 号。

【样例 3 输入输出】

见选手目录下的 ***drink/drink3.in*** 与 ***drink/drink3.ans***。

【提示】

为保证答案精度，我们一般需要尽可能地在运算过程中保留超过 p 位小数。我们可以证明，在各个子任务的参考算法中都能保证，在任何时候始终保留 $\frac{6}{5}p$ 位小数时，对任何输入得到的输出，与参考答案的绝对误差都小于 10^{-p}。

为了方便选手处理高精度小数，我们提供了**定点高精度小数类**。选手可以根据自己的需要参考与使用该类，也可以不使用该类。其具体的使用方法请参考下发的文档 ***drink/decimal.pdf***。

请注意，最终你提交的文件仅为 ***drink.cpp/c/pas***，对其它任何文件的修改均不算作有效的作答。

【子任务】

所有测试数据的范围和特点如下表所示：

<table>
<tr><th>测试点编号</th><th>n</th><th>k</th><th>p</th></tr>
<tr><td>1</td><td>≤ 2</td><td rowspan="3">≤ 5</td><td rowspan="8">$= 5$</td></tr>
<tr><td>2</td><td>≤ 4</td></tr>
<tr><td>3</td><td>≤ 4</td></tr>
<tr><td>4</td><td rowspan="4">≤ 10</td><td>$= 1$</td></tr>
<tr><td>5</td><td>$= 10^9$</td></tr>
<tr><td>6</td><td rowspan="2">≤ 10</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td><td rowspan="4">≤ 100</td><td>$= 1$</td></tr>
<tr><td>9</td><td>$= 10^9$</td><td rowspan="4">$= 40$</td></tr>
<tr><td>10</td><td rowspan="8">$\leq 10^9$</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td><td>≤ 250</td><td>$= 100$</td></tr>
<tr><td>14</td><td>≤ 500</td><td>$= 200$</td></tr>
<tr><td>15</td><td rowspan="3">≤ 700</td><td rowspan="3">$= 300$</td></tr>
<tr><td>16</td></tr>
<tr><td>17</td></tr>
<tr><td>18</td><td>≤ 2500</td><td>$= 1000$</td></tr>
<tr><td>19</td><td>≤ 4000</td><td>$= 1500$</td></tr>
<tr><td>20</td><td>≤ 8000</td><td>$= 3000$</td></tr>
</table>

旷野大计算

【问题描述】

随着人类计算机技术的发展，计算机的能力不断提升，让跳蚤国王非常羡慕。终于有一天，跳蚤国王发布政令：大力发展跳蚤国的计算机产业！

然而，跳蚤国尚未进行工业革命，无法制造出电子计算机所需的元器件。但是跳蚤国王想出了一个绝妙的想法：把每只跳蚤作为一个计算节点，每只跳蚤只完成一个特定的小任务。

跳蚤国王带领 n 只跳蚤来到了一片旷野上，把跳蚤作为计算节点在旷野上排列好，并编号为 1 到 n。每个计算节点会把某几个（也有可能是 0 个）计算节点的结果作为输入，计算得到输出。除此之外，跳蚤国王还有一个巨型的终端，可以从终端输入和输出数据，这台终端和所有计算节点组成了一台计算机。

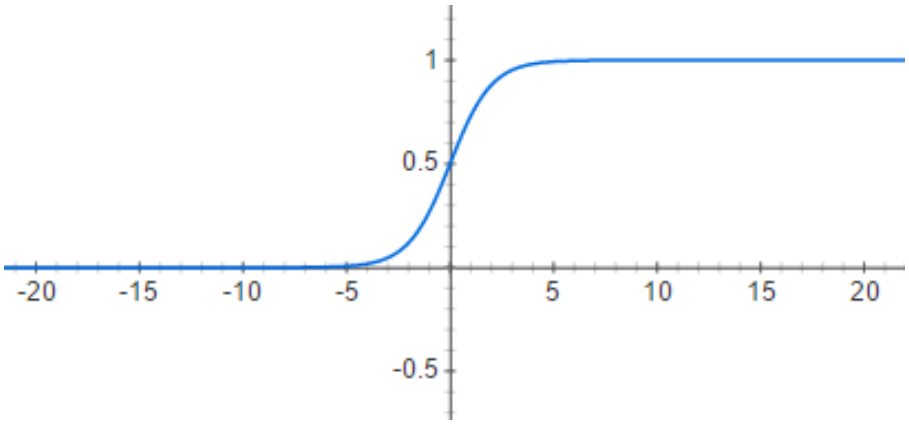
记第 t 个计算节点的输出为 x_t，该节点的操作可分为以下几种类型：

名称	操作符（类型）	操作数	计算结果
输入节点	I	无	从终端读入一个实数作为 x_t
输出节点	O	i	$x_t = x_i$，并将 x_t 输出到终端
加法节点	+	i,j	$x_t = x_i + x_j$
偏移节点	C	i,c	$x_t = x_i + c$
取反节点	-	i	$x_t = -x_i$
左移节点	<	i,k	$x_t = x_i \cdot 2^k$
右移节点	>	i,k	$x_t = x_i / 2^k$
S 型节点	S	i	$x_t = s(x_i)$
比较节点	P	i,j	$x_t = \begin{cases} -1 & x_i < x_j \\ 0 & x_i = x_j \\ 1 & x_i > x_j \end{cases}$
Max 节点	M	i,j	$x_t = \begin{cases} x_i & x_i > x_j \\ x_j & x_i \le x_j \end{cases}$
乘法节点	*	i,j	$x_t = x_i \cdot x_j$

其中，$s(x)$ 的定义如下：（e 为自然常数，其值约为 2.718281828459045 …）

$$s(x) = \frac{1}{1 + e^{-x}}$$

$s(x)$ 的函数图像如下图所示：



上述表格中的操作数 i,j 均要小于当前节点的编号 t，这样随着跳蚤国王的一声令下，跳蚤就可以按编号从小到大的顺序，依次获得输入然后计算输出。每个跳蚤的计算能力都是有限的，他们仅可以精确到十进制小数点后 90 位，超过的部分将会被四舍五入。同理，上述表格中的操作数 c 的小数部分也不能超过 90 位。另外，左移节点和右移节点中的操作数 k 必须是非负整数，且不能超过 10000。

把跳蚤排列好后，野心勃勃的跳蚤国王决心测试一下这台由跳蚤组成的计算机的计算能力，于是蝈蝈大臣给跳蚤国王献上了 10 个计算任务。完成每个计算任务均需要从终端获取输入，进行中间计算，再用输出节点将结果输出。具体任务说明如下：

编号	输入	输入限制	输出
1	a, b	$\|a\|, \|b\| \le 10^9$ 小数部分不超过 9 位	$-2a - 2b$
2	a	$\|a\| \le 10^9$ 小数部分不超过 9 位	$\frac{1}{1+e^{17a}}$
3	a	$\|a\| \le 10^9$ 小数部分不超过 9 位	$\begin{cases} -1 & a < 0 \\ 0 & a = 0 \\ 1 & a > 0 \end{cases}$
4	a	$\|a\| \le 10^9$ 小数部分不超过 9 位	$\|a\|$，即 a 的绝对值
5	$a_1, \ldots, a_{32}$	$a_1, \ldots, a_{32} \in \{0, 1\}$	把 $a_1, \ldots, a_{32}$ 从左到右看成一个二进制整数，高位在左低位在右，输出该整数的值
6	a	$0 \le a < 2^{32}$ a 为整数	输出 32 个整数，从高位到低位输出 a 的二进制表示（不足 32 位的在高位补 0）
7	a, b	$0 \le a, b < 2^{32}$ a, b 均为整数	a, b 按位异或的结果
8	a	$\|a\| \le 10^9$ 小数部分不超过 9 位	$\frac{a}{10}$
9	$a_1, \ldots, a_{16}$	$\|a_1\|, \ldots, \|a_{16}\| \le 10^9$ 小数部分不超过 9 位	输出 16 个实数，表示 $a_1, \ldots, a_{16}$ 从小到大排序后的结果
10	a, b, m	$0 \le a, b < 2^{32}$ $1 \le m < 2^{32}$ a, b, m 均为整数	$a \cdot b$ 除以 m 的余数

跳蚤国王发现自己没有足够的能力设计这样的计算机。于是他找到了来参加 NOI 的你。请你依次设计每个计算节点的类型及操作数，完成蝈蝈大臣给的这 10 个计算任务，且要求使用的计算节点数尽量少。

【输入格式】

输入文件 ***nodes1.in~nodes10.in*** 已在试题目录下，分别对应 10 个计算任务。

每组输入数据仅包含一个整数，表示需要解决的计算任务编号。

【输出格式】

输出文件为 ***nodes1.out~nodes10.out***，分别对应相应的输入文件。

对于每组输入数据，你需要依次输出若干行，第 i 行描述第 i 个计算节点。

描述每个计算节点时，首先一个字符表示该计算节点的类型，接下来若干个数按顺序表示该计算节点的内置参数。字符与数，数与数之间均用空格隔开。

输出的行数不能超过 10^4 行。

【样例输入】

```
1
```

【样例输出】

```
I
+ 1 1
- 2
I
+ 4 4
- 5
+ 3 6
- 7
- 8
O 9
```

【样例说明】

该样例输出为第一个计算任务一个可能的构造。共用了 10 个计算节点，可获得 3 分。

【子任务及部分分】

我们提供了十个评分文件 ***nodes1.ans~nodes10.ans***，分别对应每个计算任务。

每个评分文件共 10 行，第 i 行一个评分参数 w_i，具体意义将在下面给出。

本题中，每个测试点单独进行评分，每个测试点 10 分。

如果选手的输出格式不合法或者参数不符合题目约定，则得 0 分。

否则，按照以下规则判定选手的输出是否正确：

首先测评器会生成若干组输入数据，并将输入数据代入你构造的计算机。

如果在代入某一组输入数据时：你构造的计算机的计算过程中，某个计算节点的计算结果的绝对值超过 10^{1000}，则得 0 分；你构造的计算机的输出中的某个值与预期的输出值相差超过 10^{-9}，则认为你的输出不正确，得 0 分。

否则，我们认为你的计算机能完成给定的计算任务，并按照以下规则得分。

对于每个测试点，我们设置了 10 个评分参数 $w_1, w_2, w_3, \ldots, w_9, w_{10}$。

假设共使用了 n 个计算节点，你的分数将会由下表给出：

得分	条件	得分	条件
10	$n \le w_{10}$	5	$n \le w_5$
9	$n \le w_9$	4	$n \le w_4$
8	$n \le w_8$	3	$n \le w_3$
7	$n \le w_7$	2	$n \le w_2$
6	$n \le w_6$	1	$n \le w_1$

若不符合表中所有条件，得 0 分；若符合表中的多个条件，则取分数最高的。

除此之外，使用比较节点、Max 节点和乘法节点的代价是极为昂贵的。因此，这三种节点每使用一种，就会从你这个测试点的得分中倒扣 4 分。

注意这里是按使用节点的种类数计算扣分，与使用次数无关。例如多次使用比较节点，只会扣除 4 分；又如同时使用了比较节点和乘法节点，即使各只使用了一次，也会扣除 8 分。

一个测试点至多被扣到 0 分，即使分数不够扣除，也不会出现负数。

【如何测试你的输出】

在终端中先切换到该试题的目录下

```
cd nodes
```

我们提供 ***checker*** 这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是，在终端中运行

```
./checker <case_no>
```

其中 `<case_no>` 是测试数据的编号。例如

```
./checker 3
```

将测试 ***nodes3.out*** 是否可以接受。

在你调用这个程序后，***checker*** 将根据你给出的输出文件给出测试的结果，其中包括：

1. 异常退出：未知错误
2. 输入文件错误：会带有输入文件错误信息，在不修改输入文件的情况下不会触发。
3. 输出错误：错误信息。

   （如果输出格式错误则不一定得到正确的错误信息）
4. `The total number of nodes is` n`. score:` x：输出可接受，你使用了 n 个计算节点，在这个测试点获得的分数为 x。

另外，你还可以在终端中使用命令

```
./checker -f <file_name>
```

来运行 `<file_name>` 表示的计算机，并通过终端进行交互。

注意：***checker*** **测试你构造的计算机时，使用的数据跟最终测试时可能不同**。

【提示】

请妥善保存输入文件 ***nodes1.in~nodes10.in*** 评分文件 ***nodes1.ans~nodes10.ans*** 和输出文件 ***nodes1.out~nodes10.out***，及时备份，以免误删。

通过自行修改输入文件和评分文件而获得的得分是无效的。