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。

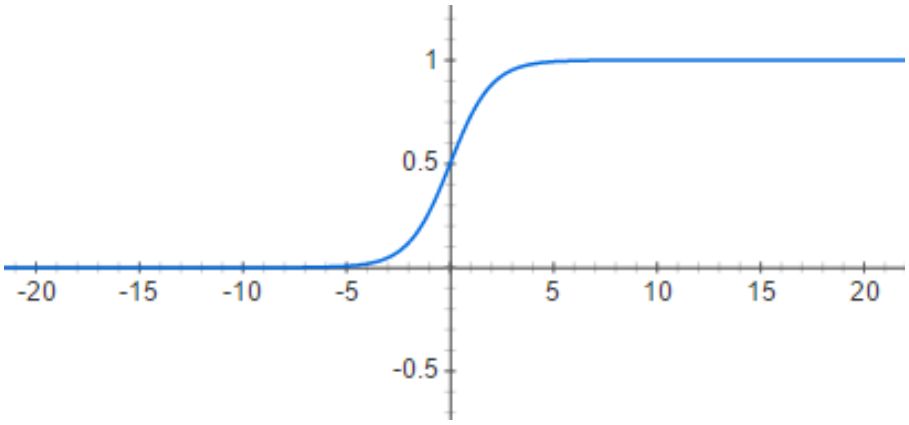
记第 t 个计算节点的输出为 x_t，该节点的操作可分为以下几种类型：

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。