

尺规作图

【问题描述】

来自软件工程系的小 W 正在工程制图的考场上，而他只带了一把被岁月磨光了刻度的直尺和一副圆规。百般无奈的他只能求助于有着惊人几何作图能力的你。现在请你使用不带刻度的直尺和圆规帮助小 W 又快又精确地完成制图。

我们来简化一下这个问题，假设小 W 已经得到了坐标系及单位长度，他需要你尺规作出长度 L_0 ，即得到点 $(L_0, 0)$ 。

你可以使用以下三种操作：

1. $line(A, B)$ ：作过点 A 与点 B 的直线；
2. $circle(O, A, B)$ ：以 AB 间距离为半径，以点 O 为圆心画圆；
3. $intersect(i, j)$ ：标记第 i 个图形与第 j 个图形的所有交点，有多个则标记多个。

初始时小 W 为你提供原点 $O(0,0)$ ，为 0 号点。单位长度由 1 号点 $(1,0)$ 给出。此外提供 x 轴（0 号图形）和 y 轴（1 号图形）。

所有在操作中得到的新的点和图形会接下去标号。

设当前标号为 n ，则 $intersect$ 操作产生的点会被标记为 $n + 1$ 。

若两图形没有交点则不会产生新的点。

若有两个交点，则 x 值较小（ x 值相同时 y 较小的）的会被标记为 $n + 1$ ，另一个为 $n + 2$ 。

请你得到点 $(L_0, 0)$ ，并指明其编号。你不需要得到精确值，但你与精确值的差距以及你所使用的操作数与你的得分相关。

【输入格式】

该题为提交答案型试题，所有输入数据 $draw1.in \sim draw10.in$ 已在试题目录下。

每一个输入数据 x_0 为一个 $python$ 的计算表达式。其中 $**$ 表示幂次运算，如

$3 ** 2$ 表示 3^2 ， $2 ** 0.5$ 表示 $\sqrt{2}$ 。

为提供较高精度，表达式中的数转为 $decimal$ 类型，即高精度小数。其中 $Decimal(2)$ 就表示 2。

输入保证数与符号，符号与符号之间会用一个空格间隔。

$python$ 代码的输出为要求构造的数。

注意直接用 $python$ 计算得到的结果是近似值，而实际要求的是精确值，即与选手获得结果进行比较的值是精确值。

另外，我们还提供了一个文件 $draw.pdf$ ，用适宜阅读的方式给出了每组数据的表达式。

【输出格式】

针对给定的 10 个输入文件 $draw1.in \sim draw10.in$ ，你需要分别提交你的输出文件 $draw1.out \sim draw10.out$ 。

输出文件的第一行一个整数 n ，表示操作数。接下来 n 行，每行一条操作，三种操作对应格式为：

```
line: L A B
circle: C O A B
intersect: I i j
```

参数意义如题目描述所述。

在输出文件的最后一行，有一个整数表示最终获得的点的编号。

【样例输入】

```
Decimal( 2 ) ** ( Decimal( 1 ) / Decimal( 2 ) )
```

【样例输出】

```
5
C 0 0 1
I 1 2
C 0 1 3
L 1 3
I 0 3
5
```

【样例说明】

样例中的五个操作意义如下：

1. 以 O 点为圆心，单位长度为半径画圆，该圆为 2 号图形；
2. 取 2 号图形（圆）与 y 轴交点，交点编号 2(0, -1), 3(0,1)；
3. 以 O 点为圆心，以 1 号点与 3 号点的距离为半径画圆，该圆为 3 号图形；
4. 连结点 (1,0), (0,1)，生成直线为 4 号图形。
5. 取 3 号图形（圆）与 x 轴交点，交点编号 4($-\sqrt{2}$, 0), 5($\sqrt{2}$, 0)。

因此 5 号节点就是最终需要得到的节点。

【评分方法】

针对给定的 10 个输入文件 *draw1.in~draw10.in*，我们分别提供评分参数文件 *draw1.ans~draw10.ans*。

评分参数文件共十行，每行两个数 c_i, x_i ，表示若最终得到的点 $(x, 0)$ 的横坐标 x 与 L_0 的绝对值之差不能超过 10^{-x_i} ，且使用操作数 $n \leq c_i$ ，则可得 i 分。

评测将取所有满足条件的最高分，若不满足任何条件得 0 分。

【如何测试你的输出】

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，首先进入终端，在终端中运行下面的命令进入本题的文件夹

```
cd draw
```

然后运行：

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

其中 *case_no* 是测试数据的编号。例如：

```
./checker 3
```

将测试 *draw3.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. 非法退出: 未知错误
2. Invalid parameters for checker.: 传递给 *checker* 的参数不合法。
3. Input/Output file does not exist.: 输入/输出文件不存在。
4. Output invalid: 输出文件有误, 此时可能包含具体错误信息。
5. Correct! Your answer is n. Your score is s.: 输出可接受, 使用的操作数为 *n*, 该测试点得分为 *s*。

【更多功能】

checker 还可以检查任意输入输出文件的测试结果, 方法是在终端中运行:

```
cd draw
```

```
./checker <input_file> <output_file> <ans_file>
```

其中 *input_file*, *output_file* 和 *ans_file* 分别是输入输出和评分参考文件的名称。

例如

```
./checker draw3.in draw3.out draw3.ans
```

将测试 *draw3.out* 是否可以接受。

使用 “-w” 可以输出生成的每个点的坐标。输出中每行格式为

```
id x y
```

分别表示点的编号, *x* 坐标, *y* 坐标。具体用法是:

```
./checker -w <input_file> <output_file> <ans_file>
```

或者

```
./checker -w <case_no>
```

例如

```
./checker -w draw3.in draw3.out draw3.ans
```

使用 “-i” 可以同时输出选手输出, 方便调试。具体用法是:

```
./checker -i <input_file> <output_file> <ans_file>
```

或者

```
./checker -i <case_no>
```

例如

```
./checker -i draw3.in draw3.out draw3.ans
```

使用 “-e *prec*” 可以指定 *checker* 运算时的精度 (有效位数)。不使用该选项时默认 *prec* 为 200。具体用法是:

```
./checker -e <prec> <input_file> <output_file>
<ans_file>
```

或者

```
./checker -e <prec> <case_no>
```

例如

```
./checker -e 50 draw3.in draw3.out draw3.ans
```

“-e *prec*”, “-w” 与 “-i” 选项可同时添加, 顺序不限, 如

```
./checker -w -i -e 50 draw3.in draw3.out draw3.ans
```

【提示】

我们可以利用 *python* 得到表达式的近似值，具体的，在终端中运行 *python* 之后，输入：

```
from math import *
from decimal import *
getcontext().prec = 100
```

然后将输入文件中的表达式（*python* 代码）复制到该终端窗口中并回车，即可得到表达式的近似值（上述代码中的 100 可替换为你想得到的精度）。

【特别提示】

如果选手使用自己生成输入文件进行调试，有可能因规模过大造成 *checker* 出错。若发生这类情况，请尝试较小规模的数据。

请妥善保存输入文件 *draw*.in*、你的输出文件 *draw*.out* 和评分参数文件 *draw*.ans*，及时备份，以免误删。