

## 问题描述

俄罗斯方块 (Tetris) 作为风靡全球 38 年的现象级益智游戏，其简单易学但难于精通的特性使其成为游戏史上的不朽经典。以下是其核心游戏规则解析及我们的要求：

游戏界面由 20 行  $\times$  10 列的可视区域组成，7 种不同形状的四格方块 (I/T/O/J/L/S/Z 型) 随机从顶部生成下落 (具体形状如下图所示)，并且在一个方块生成后，玩家可以知道下一个方块的形状。



- 横向范围：X 轴区间  $[0, 9]$  (共 10 列)
- 纵向范围：Y 轴区间  $[0, 19]$  (共 20 行， $Y=0$  为底部， $Y=19$  为顶部)
- 死亡判定线： $Y \geq 15$  (当方块堆叠高度触达此线时游戏结束)

在本问题中玩家使用的是简化版的操作规则，只能进行以下三种操作：

- 横向移动：左右平移
- 旋转操作：支持  $0^\circ$ 、 $90^\circ$ 、 $180^\circ$ 、 $270^\circ$  四向旋转
- 瞬间落地：瞬间抵达当前可到达的最底部位置

当一行及以上 (Y 轴坐标相同) 的全部位置被完全填充时，这些行将被消除，并且玩家将根据同时消除的行数获得分数：

- 单行消除：100 分
- 双行消除：300 分
- 三行消除：500 分
- 四行消除：800 分

本次作业要求做两个版本，一个 OJ 版，一个图形化版。其中 OJ 版需个人独立完成，图形化版需由 3 人组成的小组合作完成。

图形化版界面及玩法自行设计，但至少要完成单人游戏及记分功能。

OJ 版为交互式题目。要求在每次输出数据后、读入下一个数据前使用 `fflush(stdout)` 刷新输出缓冲区。

## OJ 版输入

第一行为两个单字符，即  $S_0$  和  $S_1$  ( $S_0, S_1 \in \{I, T, O, J, L, S, Z\}$ )。 $S_0$  和  $S_1$  分别表示第一个要下落的方块和第二个要下落的方块的形状。交互过程中，你每次依次输出一个方块的放置方案。在你每次输出放置方块的方案后，交互程序将再给你一行新的输入，该输入为一个单字符，即  $S_i$  ( $S_i \in \{I, T, O, J, L, S, Z, X, E\}$ )。表示一个

新的下个要下落的方块的形状。若  $S_i$  为 X 表示 X 前边的方块是最后一个方块，程序应该在输出最后一个方块的放置方案后结束。若  $S_i$  为 E 表示玩家的放置方案已导致游戏结束（堆叠高度触达死亡判定线），程序应直接退出。测试数据均采用随机种子生成，且保证输入数据的总长度（即掉落的方块总数）小于 1,000,000。

## 输出

每次交互需输出两行，第一行为用一个空格分隔的两个整数，依次为 c 和 x。其中 c 表示方块顺时针旋转的角度（0、90、180、270），x 表示方块在 x 坐标上的最小值。第二行为一个整数，表示放置完方块后玩家当前的得分。

## 一次交互过程示例

交互程序会首先提供给你 1 行 2 个输入：

IT

表示第一个方块为 I，下一个方块为 T。

然后你的程序输出将方块 I 旋转的角度和放置的位置及当前玩家得分，比如：

0 0

0

表示顺时针旋转 0 度、最左端放置在 x=0 的列上，当前得分为 0。

然后交互程序提供新的下一个方块的形状：

T

此时当前方块形状变为 T、下一个方块的形状也为 T。

你的程序在输出将方块 T 旋转的角度和放置的位置及当前玩家得分，比如：

0 4

0

然后交互程序提供输入 X 告诉你方块已经终止：

X

最后你输出最后一个 T 的放置方式及当前玩家得分，比如：

0 7

100

此时最下面一行刚好被放满，因此玩家当前得分为 100 分。

## OJ 版评分规则

假设：1、OJ 版满分为 100 分，其中基础分 90 分，奖励分 10 分；2、你的程序在学术诚信、代码风格等项上没有被扣分。

如果你的程序中的玩家得分大于等于我们设定的基础分数（这个基础分数待确定），则可以得到基础分 90 分；如果小于我们设定的基础分数，则 OJ 版成绩为 0 分（将不参与奖励分排名）。

我们会将全年级同学中所有玩家得分大于等于我们设定的基础分数的程序根据玩家得分进行排名。排名前 5% 的程序得奖励分 10 分；排名前 5% 到 15% 的程序得奖励分 8 分；排名前 15% 到 30% 的程序得奖励分 6 分；排名前 30% 到 50% 的程序得奖励分 4 分；排名前 50% 到 75% 的程序得奖励分 2 分。注意：因为直接计算每档的人数未必是整数，且可能有同分的程序（同分的程序得分一定相同），所以同学们不要死扣这个比例，这不是一个精准的范围。

提交分两次。第一次为验证数据，是给大家调试程序用的，不限提交次数。你的程序基于验证数据的分数与 OJ 版最终得分无关；第二次为测试数据，OJ 版最终得分由你的程序在测试数据上的表现决定。注意：在测试数据上我们允许的正常提交次数为 3 次。超过 3 次之后每多提交一次扣 10 分，扣到 OJ 版得 0 分为止。

说明：测试数据和验证数据基于相同的方式生成，但会使用不同的随机种子。