

【赛题】2017 年 TEG 乐趣编程对抗赛 —— 俄罗斯方块之终极对决

比赛形式：分组循环，交叉淘汰制。

第一部分【题目内容】

1) 简述

在这次的比赛中，你将和对手进行一场俄罗斯方块的游戏对决，不同的是，进行对决的是你们的程序。在游戏过程中，你可以指定对手的方块出现序列，同时，也要负责移动、摆放自己的俄罗斯方块。此外，如果你觉得当前的局势不妙，你还可以花费一定的积分，把当前的方块替换成任意一个你想要的方块。最终，分数高者获胜。

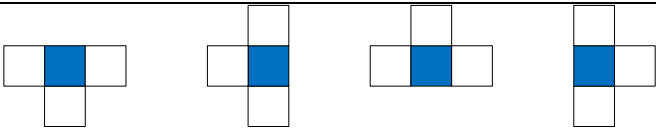
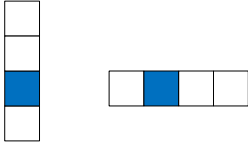

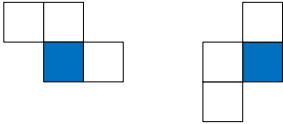
2) 你需要写两个程序。两个程序应该能够分别完成以下操作：


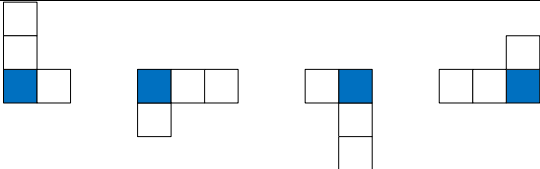
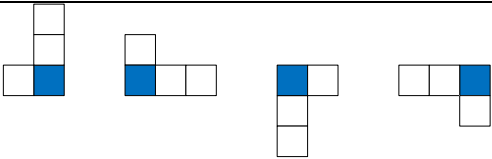
第一个程序【摆放】：用来摆放自己区域的俄罗斯方块以争取更高的分数。当方块落下且固定时，所有被填满的行将发生消除，并获得相应积分。

第二个程序【障碍】：用来决定对手区域方块出现的序列，以阻碍对手获得积分。

- 摆放区域大小：
宽度：12，高度：30。
- 方块说明：

比赛中总共有七种方块，分别用大写英文字母：T、I、S、Z、O、L、J 表示：

方块类型	图示	旋转形态数
T 型方块		4 种形态
I 型方块		2 种形态
S 型方块		2 种形态
Z 型方块		2 种形态

O 型方块		1 种形态
L 型方块		4 种形态
J 型方块		4 种形态

说明：蓝色的格子，表示为该图形的**焦点位置**，数据中方块的位置描述的一律是焦点位置。

- 新方块出现位置与形状：

新的方块总是图示中的**第一个形态**出现。并且其焦点格（图示蓝色格子）一定出现在最顶层的第 6 列（从第 1 列开始算，即中心线的左边一格）。方块上面的部分有可能超出游戏区域，这种情况是允许的。

- 旋转规则：

- 旋转时，空间必须可以容纳旋转之后的形状，否则旋转无效；比如方块因为贴着侧边或者贴着其他方块等原因，旋转后会导致方块冲突，或者超出边界（上边界除外），则系统会判定为旋转失败。
- 旋转时，焦点位置保持不变；
- 对于 1 种形态的方块，将忽略任何旋转命令；
- 对于 2 种形态的方块，无论是逆时针或顺时针，只会在 2 种状态中进行切换；
- 对于 4 种形态的方块，可以选择逆时针或者顺时针旋转。
- 旋转的判定为**逐次判定**，比如顺时针旋转 3 次，需要同时保证顺时针旋转 1 次、2 次、3 次都是合法的才能全部成功，**而不是只保证最后状态合法**。
- 旋转失败时，方块会保持最后一次成功旋转的状态。比如指令为顺时针旋转 2 次，但实际只能成功旋转 1 次。那么实际效果就是顺时针旋转 1 次。
- 特别说明：旋转的时候**上边界允许越界**。

- 移动规则：

- 可以左移，或者右移。
- 移动时碰到障碍或者边界时无效，将忽略多余的移动命令，保持最后一次成功的移动。比如左移 5 格，但实际只能移动 3 格，那么实际效果就是左移 3 格。

- 购买规则：

每回合开始的时候，你的【摆放】程序可以花费 20 积分指定购买七种方块中的任意一种，用以替换掉当前回合的方块。积分不足也可以购买，积分可扣为负数。

3) 计分说明

当方块落下且固定时，所有被填满的行将发生消除，并获得积分。每次消除的行数都会获得相应的积分。

具体如下：

- A. 每次消除1行，获得10分。
- B. 每次消除2行，获得30分。
- C. 每次消除3行，获得60分。
- D. 每次消除4行，获得100分。
- E. 每次购买方块，扣掉20分。

4) 判分说明

以你在游戏中获得的分数高低来决定胜负。其它情况说明：

- a) 如果双方的分数相等，则进行回合数更多的一方获胜。倘若回合数也正好相等，则以双方程序运行总时间来进行比较，耗时更少的一方获胜。评测程序会记录每回合双方程序的处理耗时。
- b) 禁止以任何非正常手段干扰对手程序或评测程序的运行，否则视为触犯正直的原则，直接判负。

5) 输入输出要求

所有数据通信，按照指定格式，**通过标准输入读入，并通过标准输出进行输出**。需要注意的是，不要对输出做缓冲，以避免产生超时的情况（参考后面的“其它判定说明”）。

A. 【摆放】程序规则

评测程序每回合给出当前分数、当前方块、下回合方块、以及当前的局面到程序的**标准输入**。同时，从程序的**标准输出**读取指令序列。

● **输入格式：**

输入采用文本格式，一共输入 32 行（以\n 换行）。第一行为当前分数，第二行为两个大写字母，第一个字母为本次要进行摆放的方块，第二个字母为下一次要进行摆放的方块。接下来 30 行为当前的游戏局面，每一行有 12 列，用小数点表示该格子为空，用#表示该格子已被占用。

例如：

[illegible]

```
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
##.....
.#...#....
.#...#####.
.#.#####.#.
.###.##..#.#
```

● 输出格式

每回合输出一组指令序列，每个指令序列占用一行，指令之间用英文逗号分隔。

指令	说明	示例
L{N}	往左移动 N 格,N 为 1-11	L10 -- 往左移动 10 格
R{N}	往右移动 N 格,N 为 1-11	R6 -- 往右移动 6 格
C{N}	顺时针旋转 N 次, N 为 1-3	C1 -- 顺时针旋转 1 次
A{N}	逆时针旋转 N 次,N 为 1-3	A1 -- 逆时针旋转 1 次
D{N}	落下 N 格,N 为 1-29	D10 -- 下落 10 格
B{N}	花费 10 积分购买一个方块， N 为大写字母：T、I、S、Z、 O、L、J。购买的方块将直 接替换掉当前回合的方块。 该指令必须位于指令序列 的开头。	BT -- 购买一个 T 形方块 用以替换当前回合的方块。

● 特别说明：如果实际上方块并不能移动（旋转）那么多格，那么以实际能移动（旋转）的次数为准。
比如，实际只能往下 5 格，但是指令为 D10，那么其效果等价于 D5。如果一个指令序列结束后，方块没有沉底，则评测系统会自动进行沉底处理。

范例：

【摆放】序列 BT,C3,R2,D5,L4,D29

B. 【障碍】程序规则

在比赛过程中，你的程序会给对手传递两个信息：1、当前方块；2、下一个出现的方块。所以每回合开始的时候，你的序列生成程序给出的是下回合的方块，当前回合的方块则是上回合指定的。特别说明：如果对手的【摆放】程序出现了购买行为指定了下一个方块，那么这种情况则会跳过你的【障碍】程序一个回合。

● 输入格式：

评测程序每回合之前会给出当前分数、当前方块和当前局面描述的文本输入(以\n分隔),输入格式如下：第一行为一个整数，表示当前分数，第二行为一个字母，表示当前方块类型(对于第一个回合，因为没有当前方块，字母会固定为小写的 n，表示没有)，接下来 30 行 12 列表示当前的局面。

例如：

```
1280
T
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
##.....
.#...#....
.#...####.
.#.####.#.
.###.##..#.#
```

● 输出格式：

输出为一行，一个大写字母，表示下一个方块是什么类型。

例如：

其它判定说明：

1. 超时判定与处罚：

双方程序每回合给出输出的时间，必须小于等于100毫秒。对于摆放程序，每超时1毫秒，扣1分。对于序列生成程序，每超时1毫秒，则对手加1分。如果单个回合超过了1秒，则直接按程序异常来进行判定。

2. 游戏结束判定：

当总共进行了一万个回合，或者中途出现新方块无法出现的情况（所需初始位置被占用），则游戏结束。进行分数结算。

3. 程序异常判定与处罚：

程序异常是指程序出现了中断，或者长时间未输出数据（超过1秒），或者数据输出格式有问题。倘若是摆放程序出现异常，那么游戏直接结束，进行分数结算。倘若障碍程序出现异常，那么则对手的分加上 $(10000 - \text{当前回合数}) * 100 / 12$ （按照剩余回合内摆放程序可能获得的最大分数来进行算分，即每12个回合最多获得100分）。

4. 资源限制：

单方面程序最大可用CPU为10个核，最大可用内存为20GB。超过则视为程序异常。

5. SVN提交格式

请把代码和可执行程序交到你的SVN下的trunk目录下的 part1和part2两个目录以及readme.txt说明文件，其中part1存放摆放程序，part2存放障碍程序，readme.txt写上使用的编码语言（见第7条限制）和其它必要的附属说明，part1/part2目录中必须包含：编译脚本(严格命名为小写make.sh)、源代码、可执行程序、运行脚本（严格命名为小写run.sh），以c++为例如part1下存放：

- make.sh（编译脚本，必须有，严格命名为make.sh）
- makefile（可选,看选手自己习惯）
- submit.cpp（源代码, 命名随意）
- submit（可执行程序, 命名随意）
- run.sh（运行脚本，必须有，严格命名为run.sh）

组委会统一使用make.sh生成可执行程序，然后使用run.sh执行程序。其中，make.sh以代码所在路径为

当前路径执行 (即在part1或part2执行./make.sh)。run.sh则以SVN的trunk目录为当前路径执行。(即在SVN的trunk目录执行./part1/run.sh和./part2/run.sh。请务必保证svn上的脚本可以完整执行并且生成二进制可执行程序，否则后果自负。

6. 输入输出限制

程序中会通过标准输入进行输入，标准输入进行输出，其中无论程序中是否使用到标准输入的内容都必须对标准输入的内容进行读取，一旦没有读取将导致读缓冲区满而无法继续写入数据，将直接判为异常。

7. 其它限制

支持Perl、Python、Go、C/C++、Java、Php七种语言，统一使用linux自带库函数，不得使用其它外部库。

8. 支持的编译版本

C/C++	4.8.5
Python	2.7.5
Java	1.8.0
Php	5.4.16
Go	1.6.3
Perl	5.16.3