名为 Threes 的游戏

(threes)

题目描述：

Alice 前不久一直着迷于由 Sirvo LLC 开发的游戏：Threes。

不过最近她已经厌烦了这个游戏。

Bob 知道，只有增加这款游戏的难度，才能让 Alice 继续玩下去。于是他在原有游戏的基础上，增加了名为“吸铁石”的元素，并重新制定了得分规则。下面，便来彻底介绍一下这个游戏。

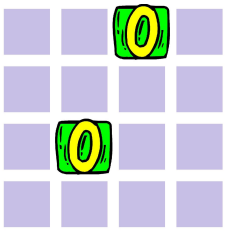

图 1-1

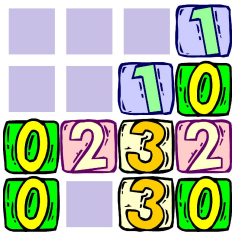

图 1-2

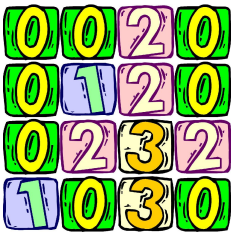

图 1-3

图 1-4

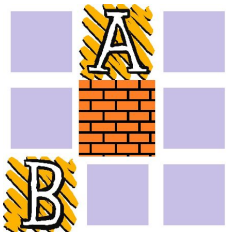

图 2-1

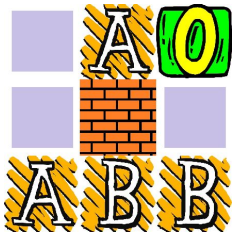

图 2-2

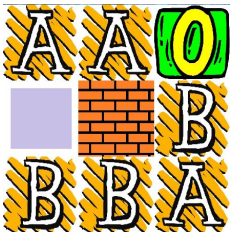

图 2-3

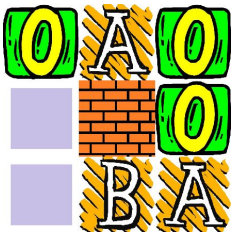

图 2-4

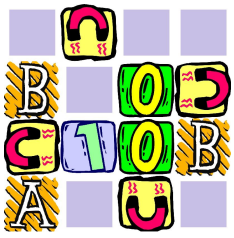

图 3-1

【一、地图】整个的游戏被设定在一个 **N 行 M 列的网格中**，每一个格子中可能出现包括障碍，吸铁石，字母，数字在内的四种符号，也可以是空白的。对于任意的当前状态，我们可以用简单的 N 行 M 列的矩阵 Map 来表示，且每一个符号对应一个分值。**当前状态的总分值为所有非空格子中符号分值的总和**。

每一个格子中有可能出现的符号我们在这里列出：

（1）非负整数。我们用数字“0”，“1”，“2”等来表示，**对于数字 n，其对应的分值为 3^(n+1)**，所以，数字 0 的分值为 3 分，数字 1 的分值为 9 分，数字 2 的分值为 27 分，以此类推。

（2）字母，只会出现大写的 A 和 B。我们用大写字母“A”和“B”来表示。**对于字母**

A，其分值为(-1)分，对于字母 B，其分值为 0 分。

（3）障碍。我们用符号“@”来表示障碍，其分值为 0。

（4）吸铁石。存在上下左右四个方向的吸铁石，我们依次用符号“^”，“v”，“<”，“>”来表示。吸铁石对于所在行所在列的格子会产生影响，但是除了所指向的方向外，对于其余三个方向来说，吸铁石可以被看作是障碍。对于所指向的方向，我们将在后面重点说明。

【二、操作】游戏每一时刻存在上下左右（我们用“W”，”S”，“A”，”D”来表示）共四种操作。对于任何的操作，网格中障碍与吸铁石的位置总不会发生变化（可以认为是被固定在网格图中的）。而字母与数字会沿着操作方向移动至不能移动，下面我们需要重点强调一下字母与数字的移动。

首先，我们需要说明：字母与数字之间有可能在碰撞后合并：

（1）对于字母来说，**A 与 B 是可以合并的**，但是 A 与 A 不能合并，B 与 B 不能合并。A 与 B **合并后会形成一个数字 0**。

（2）对于数字来说，**只有相同的数字才能合并**，不相同的数字一定不会合并。两个数字 n 合并后会得到一个数字 n+1。

每一次的操作（上下左右）都会被作用到网格图中所有的符号。对于某一行来说，向左的操作“A”（字符“A”在网格图中是一个符号，对于某个操作序列来说则是向左的操作，它们虽然公用了相同的符号“A”，但并不会造成歧义）可以认为会**依次作用**在这一行从左往右的每一个符号上（不考虑障碍与吸铁石），网格图中的状态会因此而发生变动。这一操作对于每一个符号的变动规律被概括为下列几点。

（1）对于每一个符号，在不受到吸铁石影响的基础上，如果左侧是空格，则会向左侧移动一格，之后会尝试继续向左移动。

（2）对于每一个符号，在不受到吸铁石影响的基础上，会不断向左移动。如果左侧碰到了网格图的边缘，障碍物或者吸铁石，则停止移动。如果左侧碰到了某个带有符号的格子，此时若两个格子不能合并，则停止移动，若可以合并，则会合并为同一个符号（只占据一个单位的格子），并继续尝试向左移动。

（3）如果这一行存在方向向左的吸铁石，则从吸铁石位置开始，**其左侧连续的没有空格间断的若干个符号被要求不能向左侧移动**（但是另外一方面，如果操作是向右，他们仍然有可能是可以移动并发生合并的）。

对于向左操作，如果地图中每一行从左往右每一个符号都被依次尝试过向左侧的移动，此刻考察整个网格图的**最右侧的一列格子**。如果**这一列格子中不存在空格，则游戏结束**。否则会基于某种规律去选择这一列的某个空格，并基于某种规律插入一个新的符号。

向右、向下，向上的操作，有着完全类似的规则。对于吸铁石，一种直观的理解是：吸铁石会对与之相连的连续符号产生吸引力，但是这样的吸引力并不足以改变他们的位置，但足以保证阻止它们尝试向反方向移动。

如果不能再继续操作了，则游戏结束。

【三、新插入的符号】对于确定的地图来说，还存在着插入序列 S1，S2 和决定插入位置的系数 X0，Ax，Bx，Cx，Y0，Ay，By，Cy。特别的，有的时候会有 S1=S2。

对于 N 行 M 列的地图，我们将行、列依次编号为 0、1、2 到 N-1 和 0、1、2 到 M-1。对于第一次**左右操作，X0 给出**了计划插入的位置（第 X0 行，0<=X0<N），**之后修改** X0 为 Ax*X0*X0+Bx*X0+Cx；对于第一次**上下操作，Y0 给出**了计划插入的位置（第 Y0 行，0<=Y0<M），之后修改 Y0 为 Ay*Y0*Y0+By*Y0+Cy。

对于每一次尝试插入的位置（例如尝试在第 X0 行的第一个位置插入新的符号，即为坐标(X0,0)的位置），如果所对应位置并不是空格，则会考察下一个位置（即考察(X0+1,0)的位置），特别的，如果下一个位置在格网图外了（即 X0+1=N），则会尝试考察第一个位置（即

(0,0)的位置）。举个例子来说，对于 N=10，如果当前 X0=3，则我们会依次考察(3,0)，(4,0)，(5,0)，...，(9,0)，(0,0)，(1,0)，(2,0)。如果都不空，则游戏结束。否则，第一次考察到的为空的点即为新插入符号所应该在的位置。然后，无论这个位置在哪，之后 X0=Ax*9+Bx*3+Cx。这是因为我们**只是考察了其他的点，但是并没有修改 X0**（在考察(X0+1,0)之后，我们并没有让 X0 加一）。

对于插入**序列 S1 和 S2，分别对应了上下的操作**，和左右的操作将要插入的新符号。记 S1 和 S2 的长度分别为 len1 和 len2，并从 0 开始编号。则对于第 i 次上下操作，S1 中第(i-1) mod len1 个符号就是要插入的新符号；对于第 i 次左右操作，S2 中第(i-1) mod len2 个符号就是要插入的新符号。举个例子来说，对于 S1=”AB01234”，前面 3 次上下操作将分别插入符号“A”，“B”和“0”；对于 S2=”AB”，前面 4 次左右操作将分别插入符号“A”，“B”，“A”和“B”，并依次类推下去。

【四、一些例子】这里，我们给出一些例子。图 1-1 是一个 4×4 的地图，初始时候，地图中有两个 0，其余位置都是空格。对于图 1-2 的情况，此刻若执行向下（或向上）操作，则两个数字 3 会合并为一个数字 4。对于图 1-3 来说，此刻若执行向下操作，或得到图 1-4（其中(0,2)位置出现的 1 是这一次操作后新插入的）。

对于图 2-1 是一个 3×3 的地图，中间有一个障碍物。图 2-2 中，若执行向上操作，(2,1)位置的 B 因为被障碍物阻挡，无法与(0,1)位置的 A 合并。图 2-3 中，如果执行向上操作，会得到图 2-4，其中(2,2)位置出现的 A 是新插入的符号。

对于图 3-1 是一个 4×4 的地图，其中有 4 个不同方向的吸铁石。此刻如果执行向上操作，并不会导致两个 0 合并为一个 1，因为受到向上吸铁石的影响。如果执行向左，并不会让(1,2)处的 0 向左移动，因为受到向左的吸铁石影响。对于(2,1)处的 1，虽然左侧有向右的吸铁石，上方(0,1)有向下的吸铁石，但是向下操作仍然可以改变其位置到(3,1)，因为吸铁石只能影响对应方向上与之相连的连续一段符号。

现在 Bob 制作好了游戏，并一共设计了 10 种不同的地图。

Alice 很开心，并希望可以得到尽可能高的分数。值得注意的是，因为字符 A 的分值小于零，所以作为玩家，有的时候可以为了得到更高的分数而**提前终止**游戏（即还没有因为上述一些情况而导致游戏结束就主动申请游戏结束）。

提交格式：

本题实质上是一道交互式题目，但却以传统题的格式给出。

作为交互题。我们为选手提供了 10 组不同的数据，分别为 threes1.in 到 threes10.in，这 10 组数据将在比赛中提供给所有选手。稍后将说明输入数据的格式。

对于每一组数据，我们希望选手可以在比赛中生成一个对应的输出文件（即 threes1.out，threes2.out 等）。稍后将说明输出数据的格式。

作为传统题。评测器中使用了一套**与选手在比赛中被提供的输入数据完全不同的输入文件**，其中对于每一个输入文件 threes.in 只有一个整数 t，表示这一组数据的编号（t 是 1 到 10 的某个整数）。我们希望选手实现程序 threes.c/cpp/pas，对于从 threes.in 中读入的唯一一个整数 t，输出第 t 个输出文件，例如：若 t=3，则输出 threes3.out 中的内容。

为了能让选手们更好地享受比赛，在比赛中，我们会提供应用程序 threes_admin.exe。

选手可以将 threes_admin.exe 与所以自己已经完成的输出文件放在同一个目录下（但不一定需要是全部的输出文件，即：如果只完成了部分测试点，比如说只完成了 threes1.out，threes2.out 和 threes3.out，则只需要将这 3 个文件与 threes_admin.exe 放在同一个目录下，而不需要考虑其余的 threes4.out 等文件），然后运行 threes_admin.exe，即可得到程序 threes.pas。

直接提交由 threes_admin.exe 生成的 threes.pas 即可。即便对于 c 或 c++选手，也可以只提交由 threes_admin.exe 生成的 threes.pas 程序。

输入格式：

对于 threes.in。

第一行有两个整数 N 和 M，表示地图的规模：N 行 M 列。

之后 N 行，每行 M 个符号，**中间没有多余的空格**。其中，符号“@”表示障碍物，符号“^”，“v”，“<”，“>”分别表示向上、下、左、右的四种吸铁石，大写字母“A”和“B”与数字“0”到“9”表示了所有的带有分值的可移动的符号。因为最大数字为 9，所以初始格网图中并不会出现大于 9 的数字。

之后一行，有一个整数 R，或者为 1 或者为 2。

如果 R＝1，则表示 S1＝S2，之后一行一个字符串，由 AB 和 0 到 9 组成，表示 S1，也表示了 S2。

如果 R＝2，则之后两行每行一个字符串，由 AB 和 0 到 9 组成，分别表示 S1 和 S2。

之后一行，四个整数由空格隔开，分别是 X0，Ax，Bx，Cx。

之后一行，四个整数由空格隔开，分别是 Y0，Ay，By，Cy。

输出格式：

只有一行，由一个只包含“W”，“A”，“S”，“D”的字符串（一次表示向上、向左、向下和向右的操作）组成，表示选手给出的操作序列。

得分判定：

对于每一个测试点，我们提供了 10 个评分系数 p(1)到 p(10)。只要选手的操作序列可以得到至少 p(u)的得分，这一测试点便可以得到至少 u 分。每一个测试点最高得分不超过 10 分。

评测的时候，会针对选手给出的操作序列进行游戏模拟，直到**操作序列结束，或游戏结束**。所以过多的额外操作并不会使操作序列得不到应有的分数。

其他：

本题在比赛中提供可操作环境界面 Game.jar，其中提供了所有 10 组数据的可操作界面。

选手的所有操作会被以对应输出文件的形式**动态保存下来**（可以在 Game.jar 同一目录下找到）。P 键可以选择强制结束游戏并退出到 Game 的主页面，此时 Game.jar 的目录下便可以找到输出文件。Game.jar 内嵌有游戏结束判定系统。

同时，在 Game.jar 相同目录下创建的输出文件，可以利用 Game.jar 中的 Loading Records 功能去查看游戏**自动模拟**过程，可以用来检验你所得到的输出文件。

在 Game.jar 中还提供了第 11 组数据，是与某些最近在社交网站中流行的游戏相同的游戏问题。选手可以**通过这一个数据去更好地直观了解游戏规则**。但是第 11 组数据的任何解答都不会得到任何额外的分数。

若在比赛中，所用机器中没有提供 java，或出现致命的系统崩溃，可尝试使用比赛中所提供的 java 安装包，以保证 Game.jar 可以顺利使用。

样例输入：

```
2 2
.0
..
1
1
0
2 1 0 1
2 1 0 1
```

样例输出 1：

```
ASDAA
```

样例输出 2：

```
ASWSADSDSDSWDW
```

样例说明：

对于第一个输出，只能得到 24 分。

对于第二个输出，可以得到 120 分，实际上可以证明这已经是可以得到的最优解了，但所给出的操作序列可能不是唯一的。