

NOI2014 浙江省组队选拔赛试题

第二试

题目名称	2048	取石子游戏	璀璨光华
可执行文件名	N/A	game	glitter
输入文件名	N/A	game.in	glitter.in
输出文件名	20481.out~ 204820.out	game.out	glitter.out
每个测试点时限	N/A	1	1
内存限制	N/A	256M	256M
测试点数目	20	10	10
每个测试点分值	5	10	10
是否有部分分	否	否	否
题目类型	提交答案型	传统型	传统型
是否有附加文件	有	无	无

提交源程序须加后缀

对于 C++ 语言	N/A	game.cpp	glitter.cpp
对于 C 语言	N/A	game.c	glitter.c
对于 Pascal 语言	N/A	game.pas	glitter.pas

注意：最终测试时，所有编译命令均不打开任何优化开关。

2048

【问题描述】

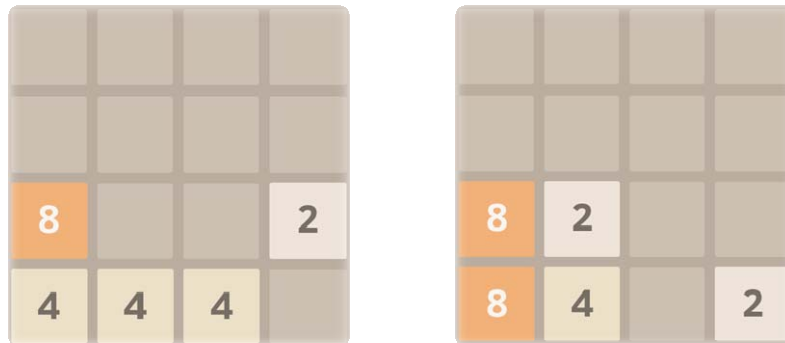
2048 是最近十分风靡的一款小游戏。

我们先介绍一下实际游戏的规则：2048 游戏由 4×4 的矩阵构成。每个位置（方格内）可以放置不多于一个瓦片(tile)，每个瓦片上标着一个值为 2 的幂的数。玩家每回合从上下左右中选择一个**合法**的方向（合法性将在稍后解释），这些瓦片将**尽可能**向该方向移动，直至边界，并且保持顺序。若相互**挤压**（即**相邻**且**相对位置平行**于所选方向）的瓦片数字相同，则立即合并为一个瓦片，消失的瓦片立即不占据位置。每回合每个瓦片**仅可合并一次**，并且合并次序以沿玩家所选方向**靠前者**为先。一个方向**合法**当且仅当该选择的方向至少移动或合并一个瓦片，因此该操作后，矩阵上至少存在一个空白位置。开始游戏前或回合间，游戏程序会在空白处随机生成一个标着 2 或 4 的瓦片。当场上合成 2048 时游戏结束，玩家胜利；若玩家不能做出合法的移动，游戏结束，玩家失败。

现在，我们想让AI尽量获得更高的分数，请注意，本题中规则与实际游戏有所不同。游戏中玩家不知道随机生成瓦片的位置和大小，在本题中，我们**给定随机算法**，并且每次只生成标着 2 的瓦片，希望依此获得相当高的分数。

游戏规则样例：

出现左图所示的状态后，玩家选择向左移动，下方的三个 4 将会变为一个 8 和一个 4，由于左边两个 4 更靠近移动方向（左），故应合成最左侧一个 8 和第二左侧一个 4（见右图），由图可以判断出，游戏程序在回合间随机生成了一个 2 在右下角：



随机算法如下：

给定随机种子 `seed`，和常量 `MUL = 8221`，一个有符号 32 位的随机整数

数列 `seq` 按如下规则生成：

其中 \gg 符号表示带符号右移。当加减法结果 (或) 时,

2

自动变为模 意义下相等大小的负数 (或正数) 【Pascal 选手可以使用 {SR-} 选项】。

2048 游戏所需的随机数为 0~15 的数字, 分别代表从左到右、从上到下的依次编号的瓦片 (见下图)。每次需要随机生成瓦片时, 我们在随机数数列中逐个取数并对 16 取模, 直至取到一个对应空当 (而非已被瓦片占据的位置) 的数为止。下次取随机数时应从上次选取数字的下一个开始。第一个随机数为 seq_1 。

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

典型的使用 C 和 Pascal 生成上述随机数序列的代码如下:

```
int MUL = 8221;
int seq;

void initSeed(int seed)
{
    seq = seed;
}

int getRand()
{
    return (seq = (seq * MUL) + (seq >> 16)) & 15;
}
```

```
Const
    MUL = 8221;
    ArraySize = 1048576;
Var
    Seq: Array[0..ArraySize] of Longint;

Procedure Init(Const Seed: Longint);
Var
    i: Longint;
Begin
    Seq[0] := Seed;
    For i := 1 to ArraySize do
        Seq[i] := (Seq[i - 1] * MUL) + (Seq[i - 1] Shr 16) + Longint(Seq[i - 1] < 0) * $FFFF0000;
    End;

Function GetRand(Const i: Longint): Longint;
Begin
    GetRand := Seq[i] and 15;
End;
```

输出文件中请按顺序包含一组合法的操作，尽可能获得标记数字更大的瓦片。评分时将根据执行完毕后的**最终状态**评定。

附加文件 `simulate(.exe)` 可以检查您的输入文件并进行模拟，输出其最终状态到控制台。

使用方法：在 terminal 或 `cmd.exe` 中调用 `simulate [filename: 2048*.out]`。

【输入格式】

本题没有输入文件，但我们在这里指定一组随机种子，第 i 题使用数值为 i 的随机种子（ $i = 1, 2, \dots, 20$ ），对应输出文件分别为 `20481.out~204820.out`。

【输出格式】

在输出文件 `20481.out~204820.out` 中，第一行输出该题目对应的随机种子（注意文件名和种子顺序不要交换）。第二行按顺序列出操作（上下左右分别用大写字母 UDLR 表示）。

【样例输入】

（无）

【样例输出】

```
1
LLLLLDLRLRDULRRDRUUDRLURLURLLUDLLURLRLLULLLLL
```

【样例说明】

以上的样例是 `20481.out` 的一个合法输出。它可能并不能得到这个测试点的满分。

【评分标准】

每个测试点单独评分，每个测试点权重相等。如果你的输出不合法或不满足要求，得分为 0，否则得该测试点满分。要求如下。

对于测试点 1-2：你的**最终状态**出现标记不小于 4096 的瓦片。

对于测试点 3-5：你的**最终状态**出现标记不小于 8192 的瓦片。

对于测试点 6-10：你的**最终状态**出现标记不小于 16384 的瓦片。

对于测试点 11-20：你的**最终状态**出现标记不小于 32768 的瓦片。

取石子游戏

【问题描述】

Roland. P. Sprague 和 Patrick. M. Grundy 都是组合游戏的狂热爱好者，但他们素未谋面。

一天，Sprague 在写给 Grundy 的信中向他介绍了一个据称是来自东方的古老游戏——取石子。

取石子是一个双人博弈游戏。在游戏的一开始，桌面上有几堆石子堆。接下来，游戏双方轮流进行操作：从桌面上选取一堆石子堆，然后从这一堆里面取走任意多个石子（但不能不取）。当某个人无法操作时则失败，另一方获得胜利。由于条件所限，Sprague 建议在纸上写一排自然数来代表各个石子堆的石子数目，然后两人轮流划数写数。Grundy 欣然应允。

一个月过去了，在 Grundy 连续输了 5 盘游戏之后，他怀疑 Sprague 耍诈。经过几天的研究，Grundy 在某天下午发现假设游戏双方都足够聪明，那么给定一个初始状态（一排自然数），可以有很简单的方法来判断先手必胜还是后手必胜，并且可以给出必胜策略！于是 Grundy 决定要进行反击。

翌日，Grundy 在写给 Sprague 的信中建议把游戏的规则改得更复杂一点：首先确定一个常数 K 。然后，游戏双方的操作改为：每次选择一个数划掉。假设该数为 x ，操作者可以任选一个正整数 a ，在划掉 x 之后需要再写上 $x \mid a, x \mid 2a, \dots, x \mid Ka$ 共 K 个数，且 a 需满足 $x \mid Ka \mid 0$ 。若这样的 a 不存在，那么操作者就不能划掉这个 x 。某一方失败的条件依然是他无法操作。

碍于面子，Sprague 当然无法拒绝。不过他也不会坐以待毙，现在他已经得到了 K 和写在纸上的 N 个数。他把这些数据 and 这个游戏的规则都告诉了你——John von Neumann——正在研究如何使用一个尚不存在的机械（你将其命名为计算机）来解决实际问题的数学、物理、经济学（、计算机科学）家。

【输入格式】

输入文件 `game.in` 第一行是一个正整数，表示数据组数 T （Sprague 向你询问的次数）。接下来依次输入 T 组数据，每组数据占 $N + 2$ 行，格式如下：

第 1 行是一个空行。

第 2 行，两个正整数，按顺序表示 N 和 K 。

接下来 N 行，每行一个正整数，表示写在纸上的 N 个数。

【输出格式】

输出文件 *game.out* 有 T 行。对每组数据，如果先手必胜（先进行操作的玩家拥有必胜策略），则输出"Preempt."。若后手必胜，则输出"Leapfrog."。若两者皆非，则输出"Je suis un imbecile."。

【样例输入】

```
2

1 1
1

2 30
197943
249832
```

【样例输出】

```
Preempt.
Leapfrog.
```

【数据规模与约定】

10%的数据满足： $N \leq 5$ ， $K = 1$ ，所有数均小等于 5。

20%的数据满足： $N \leq 100$ ， $K = 1$ ，所有数均小等于 10^9 。

10%的数据满足： $N \leq 100$ ， $K = 2$ ，所有数均小等于 10^9 。

20%的数据满足： $N \leq 100$ ， $K = 2$ ，所有数均小等于。

20%的数据满足： $N \leq 100$ ， $K = 10$ ，所有数均小等于。

40%的数据满足： $N \leq 100$ ， $K = 30$ ，所有数均小等于。

100%的数据满足： $T \leq 10$ 。

璀璨光华

【问题描述】

金先生有一个女朋友——没名字。她勤劳勇敢、智慧善良。金先生很喜欢她。为此，金先生用 a^3 块 $1 \times 1 \times 1$ 的独特的水晶制作了一个边长为 a 的水晶立方体。他要将这个水晶立方体送给他见过最单纯善良的她。

由于水晶立方体太大，不好运送，金先生还是将它拆开来送出。他相信拼好这个水晶立方体难不倒聪明的她。

没名字收到了礼物后果然不一会儿就根据说明将水晶立方体拼好了。没名字发现，有 n 块水晶在漆黑安静的夜晚会随机向上下左右前后六个方向的一个发出光。被光照到的水晶显得格外好看。没名字给每一块不会发光的水晶定义了一个好看程度。水晶立方体在夜晚中的好看程度就是每块被光照到的水晶的好看程度之和。没名字想知道，水晶立方体在夜晚中的好看程度的最小值和最大值。

【输入格式】

输入文件 `glitter.in` 的第一行是 a ，表示水晶立方体的边长。

接下来 a^3 行，每行若干整数。第一个整数 $g[i]$ ，表示第 i 块水晶的好看程度。

如果 $g[i] = 0$ ，代表这块水晶会发光。接下来 3-6 个整数，代表与这块水晶有共同面的水晶编号。

【输出格式】

输出文件 `glitter.out` 总共 1 行，两个整数，水晶立方体在夜晚中的好看程度的最小值和最大值。

【样例输入】

```
2
0 7 2 3
0 8 1 4
4 5 4 1
8 6 3 2
16 3 6 7
32 4 5 8
1 1 8 5
2 2 7 6
```

【样例输出】

0 12

【数据规模与约定】

数据点	a	n
1	3	1
2	3	5
3	3	8
4	1 10	1 6
5	1 50	1 6
6	1 70	1 6
7	1 10	1 8
8	1 30	1 8
9	1 50	1 8
10	1 70	1 8

对于 100% 的数据， $a[i] \leq 1000000, a > 1$ 。