

测试题

(请选手务必仔细阅读此页内容)

一. 题目概况

| | | | | |
|-----------|---------------|---------------|---------------|---------------|
| 中文题目名称 | 比赛 | 游戏 | 还原 | 方格 |
| 英文题目与子文件名 | match | game | restore | Board |
| 可执行文件名 | match | game | restore | Board |
| 输入文件名 | match.in | game.in | restore.in | board.in |
| 输出文件名 | match.out | game.out | restore.out | board.out |
| 每个测试点时限 | 1 秒 | 1 秒 | 1 秒 | 1 秒 |
| 内存上限 | 128MB | 128MB | 128MB | 128MB |
| 测试点数目 | 10 | 10 | 20 | 20 |
| 每个测试点分值 | 10 | 10 | 5 | 5 |
| 结果比较方式 | 全文比较 (忽略行末空格) | 全文比较 (忽略行末空格) | Special Judge | 全文比较 (忽略行末空格) |
| 题目类型 | 传统 | 传统 | 传统 | 传统 |

二. 提交源程序文件名

| | | | | |
|--------------|-----------|----------|-------------|-----------|
| 对于 Pascal 语言 | match.pas | game.pas | restore.pas | board.pas |
| 对于 C 语言 | match.c | game.c | restore.c | board.c |
| 对于 C++语言 | match.cpp | game.cpp | restore.cpp | board.cpp |

三. 编译命令

| | | | | |
|--------------|---------------------------|-------------------------|-------------------------------|---------------------------|
| 对于 Pascal 语言 | fpc match.pas | fpc game.pas | fpc restore.pas | fpc board.pas |
| 对于 C 语言 | gcc -o match match.c | gcc -o game game.c | gcc -o restore restore.c | gcc -o board board.c |
| 对于 C++语言 | g++ -o match match.cpp | g++ -o game game.cpp | g++ -o restore restore.cpp | g++ -o board board.cpp |

注意事项:

- 1、文件名 (程序名和输入输出文件名) 必须使用英文小写字母。
- 2、C/C++ 中函数 `main()` 的返回值类型必须是 `int`, 程序正常结束时的返回值必须是 0。
- 3、题目简单, 请认真对待。
- 4、四道题源代码长度限制均为 50K B。
- 5、每位选手提交一个以自己编号命名的文件夹, 在该文件夹下放4个源程序, 然后再在该文件夹下建立四个子目录, 名称分别为: `match`、`game`、`restore`、`board`, 再把源代码放入对应的子目录中。
- 6、四道题的数据都有一定梯度。请尽量优化算法, 争取拿高分。
- 7、特别提醒: 评测分别在 **NOI Linux** 下国评系统与 Windows 下 lemon 下进行。

1.比赛

(match.pas/c/cpp)

【问题描述】

果果非常喜欢看足球赛，但因为沉迷于刷集训队作业，错过了最近的一次足球联赛。此次联赛共 n 支球队参加，比赛规则如下：

- (1) 每两支球队之间踢一场比赛。
- (2) 若平局，两支球队各得 1 分。
- (3) 否则胜利的球队得 3 分，败者不得分。

尽管非常遗憾没有观赏到精彩的比赛，但 r64 通过新闻知道了每支球队的最后总得分，然后聪明的他想计算出有多少种可能的比赛过程。

譬如有 3 支球队，每支球队最后均积 3 分，那么有两种可能的情况：

| 可能性 1 | | | | | 可能性 2 | | | | |
|-------|---|---|---|----|-------|---|---|---|----|
| 球队 | A | B | C | 得分 | 球队 | A | B | C | 得分 |
| A | - | 3 | 0 | 3 | A | - | 0 | 3 | 3 |
| B | 0 | - | 3 | 3 | B | 3 | - | 0 | 3 |
| C | 3 | 0 | - | 3 | C | 0 | 3 | - | 3 |

但 r64 发现当球队较多时，计算工作量将非常大，所以这个任务就交给你了。请你计算出可能的比赛过程的数目,由于答案可能很大,你只需要输出答案对 10^9+7 取模的结果。

【输入】

第一行是一个正整数 n ，表示一共有 n 支球队。

接下来一行 n 个非负整数，依次表示各队的最后总得分。

【输出】

包含一个整数，表示答案对 10^9+7 取模的结果。

【输入输出样例】

| match.in | match.out |
|--------------|-----------|
| 4 4 3 6 4 | 3 |

【数据说明】

20%: $n \leq 4$;

40%: $n \leq 6$;

60%: $n \leq 8$;

100%: $n \leq 10$ 且至少存在一组解。

2.游戏

(game.pas/c/cpp)

【问题描述】

果果最近被安利一个新游戏，这个游戏由一个三角形盘件和 12 个形态各异的零件组成。拼盘的盘件如图 1 所示：

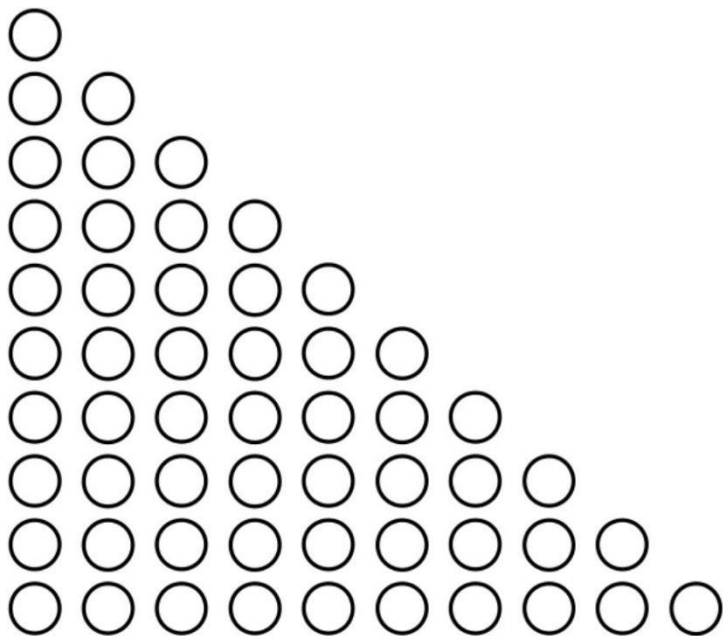
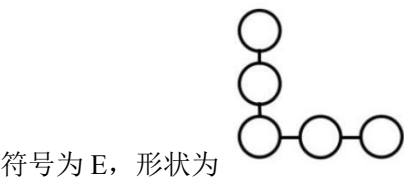
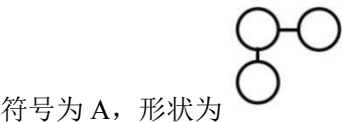
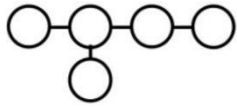


图 1

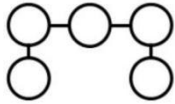
这 12 个部件为分别为：



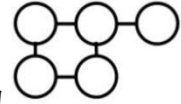
符号为 F，形状为



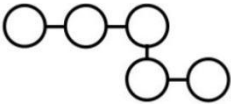
符号为 G，形状为



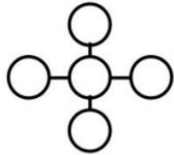
符号为 H，形状为



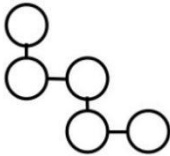
符号为 I，形状为



符号为 J，形状为



符号为 K，形状为



符号为 L，形状为

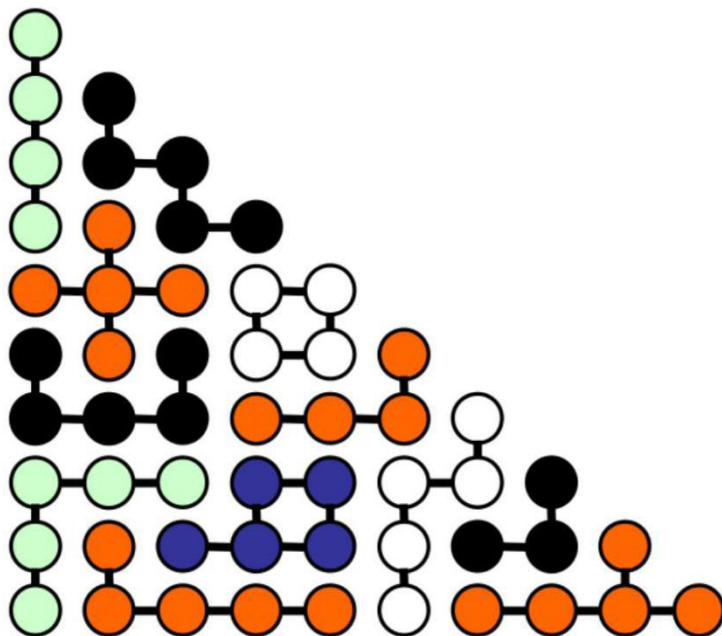
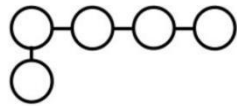


图 2

图 2 示出了一种拼盘方案。为便于描述可将图 2 抽象为图 3，就可以用一个数据为字符

的二维数组来表示了。

```

B
B  K
B  K  K
B  J  K  K
J  J  J  D  D
G  J  G  D  D  C
G  G  G  C  C  C  I
E  E  E  H  H  I  I  A
E  L  H  H  H  I  A  A  F
E  L  L  L  L  I  F  F  F  F
```

图 3

对于由珠子构成的零件，可以放到盘件的任一位置，条件是能有地方放，且尺寸合适，所有的零件都允许旋转(0°，90°、180°、270°)和翻转(水平、竖直)。

现给出一个盘件的初始布局，求一种可行的智慧珠摆放方案，使所有的零件都能放进盘件中。

【输入】

一共有 10 行,其中第 i 行有 i 个字符。

如果第 i 行的第 j 个字符是字母'A'至'L'中的一个,则表示第 i 行第 j 列的格子上已经放了零件，零件的编号为对应的字母。如果第 i 行的第 j 个字符是'.'，则表示第 i 行第 j 列的格子上没有放零件。

输入保证预放的零件已摆放在盘件中。

【输出】

如果能找到解,则输出 10 行，为放完全部 12 个零件后的布局。其中,第 i 行应包含 i 个字符,第 i 行的第 j 个字符表示第 i 行第 j 列的格子上放的是哪个零件。

如果无解，输出单独的一个字符串“No solution”。

所有的数据保证最多只有一组解。

【输入输出样例】

| game.in | game.out |
|-----------|------------|
| . | B |
| .. | BK |
| ... | BKK |
| | BJKK |
| | JJJDD |
|C | GJGDDC |
| ...CCC. | GGGCCCCI |
| EEEEH... | EEEHIIIA |
| E.HHH.... | ELHHHIAAF |
| E..... | ELLLLIFFFF |

3.还原

(restore.pas/c/cpp)

【问题描述】

在几何课上，老师画了一个圆，圆上有很多条弦，这些弦两两不重合，但是有些弦是相交的。果果本想把图临摹下来回家好好研究，可惜下课后，图被熊孩子琪琪擦掉了。幸运的是，果果准确地记录了弦的数量和弦的相交情况。

现在，果果想尽量复原这张图。同时果果还想知道，最多能选出多少条弦，使得选出来的弦两两不相交，但是他发现他并不会做，于是他把问题交给你，你能帮帮他吗？

【输入】

第一行两个正整数 n, m ，分别表示弦的条数以及相交弦的对数，所有的弦从 1 至 n 编号。

接下来一共 m 行，每行两个正整数 a, b ，表示第 a 条弦与第 b 条弦相交。

【输出】

第一行 $2n$ 个正整数，按逆时针方向给出满足题意的圆上每条弦的两个端点的相对顺序，其中第 i 条弦的两个端点均用数字 i 来表示，如果存在多组解只需任意输出一组即可。

第二行一个正整数，表示最多能选多少条两两不相交的弦。

【输入输出样例 1】

| restore.in | restore.out |
|------------|---------------------|
| 5 6 | 1 2 3 1 4 2 5 4 3 5 |
| 1 2 | 2 |
| 1 3 | |
| 2 3 | |
| 2 4 | |
| 3 5 | |
| 4 5 | |

【数据说明】

$1 \leq n \leq 20, 1 \leq m \leq 40$ ，保证至少有一组解。

4.方格

(board.pas/c/cpp)

【问题描述】

果果有一个 $n*m$ 的方格。一些格子上已经被涂上了 k 种颜色中的任意一种。果果想把剩下所有未染色的格子全部染色，每个格子都可以选 k 种颜色中的一种。使得涂完后，从方格的左上角到右下角不存在一条有两个相同颜色格子的路径，并且路径只能向下或向右走。

现在果果想知道，一共有多少种染色方法能够满足以上的条件？

【输入】

第一行一共三个整数 n ， m ， k ，分别表示方格的长和宽，以及颜色的种数。

接下来一共 n 行，每行 m 个整数。设第 i 行第 j 列的数为 a_{ij} ，如果 $a_{ij} \neq 0$ ，则表示该格子已经被染色，且颜色的编号为 a_{ij} ，如果 $a_{ij} = 0$ ，则表示该格子还未被染色。

【输出】

一个整数，表示染色方案数，并对 10^9+7 取模。

【输入输出样例 1】

| board.in | board.out |
|---------------------|-----------|
| 2 2 4 0 0 0 0 | 48 |

【数据说明】

$1 \leq n, m \leq 1000$ ， $k \leq 10$ 。