

弱弱的 WC 2015 模拟赛

zyhzyhzyh, Ruchiose, xllend3

推箱子	东舰停战不可避	kenji's life
box	tkw	kenji
2.5 s	3.5 s	(提交答案型问题)
512 MB	512 MB	(提交答案型问题)

注意：

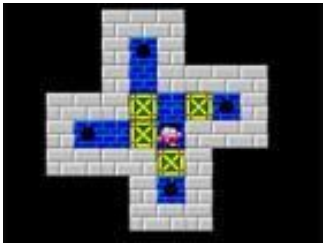
- box 这题采用捆绑测试，所以测评软件是 lemon；
- 测评使用的是 lemon，所以不需要为每个题目建立子文件夹。

推箱子

(box.cpp/c/pas, 2.5s, 512 MB)

【题目描述】

经典的推箱子是一个来自日本的古老游戏，目的是在训练你的逻辑思考能力。在一个狭小的仓库中，要求把木箱放到指定的位置，稍不小心就会出现箱子无法移动或者通道被堵住的情况，所以需要巧妙的利用有限的空间和通道，合理安排移动的次序和位置，才能顺利的完成任务。



这个游戏有一个相对简单的版本，就是只有一个木箱，要将其推到一个确定的目标位置。举个例子：

```
..#@.
.X.O.
##. .#
```

其中“.”表示没有障碍的格子，“#”表示有障碍的格子，“X”表示目标位置（注意这个格子是没有障碍的），“O”是箱子（对于人来说箱子存在的格子是不能越过的），“@”是人的位置。注意上面这个情况是可以将箱子推到目标点去的，因为人只要向右一格，向下一格，然后推着箱子向左走两格，就完成了任务。

而下面这个例子是无解的：

```
..#..
.X.O.
##.@#
```

现在我们的主人公想知道，对于一个给定的地图（与上面两个不同的是没有了箱子和人的初始位置），有多少种箱子和人的初始摆放方法（箱子的初始位置和人的初始位置和目标位置必须两两不同）能够使得箱子能被人成功地推到目标位置。

【输入格式】

第一行为两个正整数 n, m 。表示整个地图有 n 行 m 列。

接下来是一个 n 行 m 列的地图，用 n 个长为 m 的字符串表示。

【输出格式】

一行，包含一个整数，为方案总数。

【样例测试数据】

box.in	box.out
3 5 ..#.. .X... ##. .#	9
2 3 .X.. ...	0

4 7 . # . # . ## ## . # . . # X . . ## . # . . .	24
--	----

【数据规模与约定】

对于 10%的数据：n, m ≤ 10。在捆绑测试中作为一组。

对于另外 20%的数据：n, m ≤ 50。在捆绑测试中作为一组。

对于另外 10%的数据：n, m ≤ 200。在捆绑测试中作为一组。

对于另外 60%的数据：n, m ≤ 1000。在捆绑测试中分为三组。

【测评方式】

本题采用捆绑测试，在每个测试点中实行全文比对。根据数据规模与约定中指出的，本题的数据被分为 6 组。每组数据中有若干个测试点，每个测试点都有 2.5 秒的单独的时间限制。你必须通过一组中的所有测试点，才能获得该组测试点的分数。

东舰停战不可避

(tkw.cpp/c/pas, 3.5s, 512 MB)

【题目描述】

一转攻势
贴吧击沉
壕大胜利
舰队厨激怒
私服厨狂喜
运营爆破不可避
东舰LL联合确定

现在为了迎战私服的壕们，舰队厨决定停止和东方厨之间的撕逼大战。然而，为了停战首先需要解决一些历史遗留问题：

在平面上有 N 个东方厨和 M 个舰队厨，你可以把他们当成平面上的点。现在 N 个东方厨要组成一个贴吧。为此他们要在平面上连 $N-1$ 条笔直的边，把 N 个东方厨连在一起。 M 个舰队厨也要组成一个贴吧，为此他们要在平面上连 $M-1$ 条笔直的边，把 M 个舰队厨连在一起。为了防止世界被破坏，我们要求，从这 $N+M-2$ 条边中选出任意两条互异的边，它们都没有交点。请构造一个方案。

【输入格式】

第一行，两个整数 N, M 。代表东方厨和舰队厨的个数。

以下 N 行，每行两个整数。代表东方厨的位置。

以下 M 行，每行两个整数。代表舰队厨的位置。

【输出格式】

如果无解，输出一行 “touhou-kancolle war can not be avoided!”（不含引号）。

否则输出 $N+M-2$ 行：

前 $N-1$ 行，每行两个整数。代表一条连接两个东方厨的边，东方厨的编号从 1 到 N 。

后 $M-1$ 行，每行两个整数。代表一条连接两个舰队厨的边，舰队厨的编号从 1 到 M 。

【样例测试数据】

tkw.in	tkw.out
3 2	1 3
1 0	2 3
0 1	1 2
2 3	
0 0	
1 1	

【样例解释】

由于东方厨 1 和东方厨 2 之间的边会和舰队厨 1 和舰队厨 2 之间的边相交，要把东方厨 1 和东方厨 2 连在一起必须要经过东方厨 3。容易看出.out 中的方案是唯一的方案。（当然，你可以调换前两条边的位置，或是调换每行的两个数的顺序，得到的都是正确的输出。）

【数据规模与约定】

对于 30% 的数据， $N+M \leq 10$ 。

对于另外的 20% 的数据，存在两个相离的多边形，前者覆盖了所有东方厨，后者覆盖了所有舰队厨。这类数据共两个，其中一个的 $N+M$ 为 100，另一个的 $N+M$ 为 3000。

对于 100%的数据, $N, M > 1$ 。 $N+M \leq 3000$ 。坐标是不超过 10^9 的自然数。没有三点共线。
没有三点共线。因为很重要所以说两次。

【测评方式】

本题设有 Special Judge。

没有对输入格式进行检验所以输出一行 $2(N+M-2)$ 个数也可能被判为正确。

请勿爆 SPJ。

检测到在途的非传统题反应!!!
非战斗人员请尽快撤离!!!

kenji's life

(kenji*.out)

【题目描述】

kenji's life 是一个有趣的小游戏。

现在我们对这个游戏进行了转化，并给出转化后的游戏，请你最大化游戏结束后某个变量的值。

变量：在 kenji's life 中，所有变量均为整型变量，任意变量无需声明即可直接使用，并自动初始化为 0。合法的变量名应只包括 A~Z, a~z, 0~9, _，并且第一位不是数字，并且不是保留字。

语句：一条语句有如下几种形式：

A opr B: 其中 A 是变量，B 是变量或自然数， $\text{opr} \in \{=, +, -, *\}$ 。

break: 跳出当前所在的 while 循环。

return: 直接结束游戏。

语句块：包含 n 条（n 为自然数）语句的整体。（即可以为空）

表达式格式：A opr B 其中 A, B 是变量或自然数， $\text{opr} \in \{=, !=, >=, <=, >, <\}$

while 循环格式：

while

语句块

endwhile

（功能等同于 c++ 中的 while(1){...}）

switch 分支格式：

switch

case 0:

语句块

case 1:

语句块

...

case n:

语句块

endswitch

其中 $n \geq 1$

（功能等同于 c++ 中的 switch(...){case 0:...break;case 1:...break;...}

即只执行到下一个 case 或 endswitch）

if 分支格式：

if 表达式

语句块

endif

或

if 表达式

语句块

else

语句块

endif

具体格式可以参考输入数据。

输出数据需要对每一个遇到的 switch 分支输出选择的 case。

你需要使游戏结束时 **ans 变量** 尽可能地大。

【输入格式】

该题为提交答案型试题，所有输入数据 kenji1.in~kenji10.in 已在试题目录下。输入的第一行包含一个整数表示数据点编号。接下来所有的部分表示一个游戏对应的代码。

【输出格式】

针对给定的 10 个输入文件 kenji1.in~kenji10.in, 你需要分别提交你的输出文件 kenji1.out~kenji10.out。

每个文件需要输出若干行，每行一个处于 0 到当前 switch 的 case 最大值之间的数，表示所做的选择。

输出的行数需要严格等于此次游戏执行到结束时所遇到的 switch 分支个数。

【样例测试数据】

kenji0.in	kenji0.out
0 while if ans >= 3 break endif switch case 0: ans += 1 case 1: ans += 2 endswitch endwhile if ans > 3 ans -= 2 endif	0 1

【样例解释】

一共有 5 种合法的输出：“000”，“001”，“01”，“10”，“11”。其中“000”，“01”，“10”值为 3，“001”，“11”值为 2。

【测评方式】

对于每组数据，我们设置了三个评分参数 a1,a2,a3。

若你的程序输出数字有误或个数有误得 0 分。

若你的程序正常结束：

若 $ans > a1$ 得 11 分

否则若 $ans \geq a1$ 得 10 分

否则若 $ans \geq a2$ 得 6 分

否则若 $ans \geq a3$ 得 3 分

否则得 1 分

【如何测试你的输出】

我们提供 checker 这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是，首先进入终端，在终端中运行下面的命令进入本题的文件夹：

```
cd train
```

然后运行：

```
./checker <in_file> <out_file>
```

其中 in_file 是输入数据的文件名 out_file 是输出数据的文件名,例如

```
./checker kenji1.in kenji1.out
```

将测试 kenji1.out 是否可以接受。

在你调用这个程序后, checker 将根据你给出的输出文件给出测试的结果。



其中 in_file 如果不是提供的 10 个数据之一不保证 checker 不会输出奇怪的东西

【注】

复制样例的时候请保证在最后一行输入后有换行符

数据在 windows 下生成, mac 和 linux 的 checker 未经过测试, 出现问题概不负责(来打我呀)

数据并不保证最优解欢迎拿 110 分