



题目名称	迷宫花园	木偶	追风者
程序文件名	maze	puppet	chaser
输入文件名	maze.in	puppet.in	chaser.in
输出文件名	maze.out	puppet.out	chaser.out
每个测试点时限	1 秒	1 秒	5 秒
内存限制	256 MB	256 MB	256 MB
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	无	无	无
提交源程序需加后缀			
对于 Pascal 语言	maze.pas	puppet.pas	chaser.pas
对于 C 语言	maze.c	puppet.c	chaser.c
对于 C++ 语言	maze.cpp	puppet.cpp	chaser.cpp

最终测试时，所有编译命令均不打开任何优化开关。

测试环境：

Cena 0.8.1 @ Windows 7 Home Premium 64-bit Edition

C/C++ 语言使用 GCC 版本 4.6.1 进行编译。

Pascal 语言使用 FPC 版本 2.0.2 进行编译。

**P.S.:**

使用 64 位系统纯属迫不得已。

如对测试结果有异议请尽快提出，对此造成的不便请您谅解。

# Violet I

## 迷宫花园

(maze.pas/c/cpp)

### 背景

我们常常迷失在这狗血的世界里。

拼命地向着目标努力，最后却总感觉有一只看不见的手在玩弄自己。

生活就像一个舞台，自己却不知道在给谁唱着哪一出折子戏。

有时候非常羡慕那些思想者，从古希腊的智者，一直到被称为“新亚里士多德学派”的人们，在嘈杂纷乱的世界里还能一直——或者说常常——清醒地知道自己的方向，尽管命运无情多变，他们也未自暴自弃。

可是我们呢，我们在做些什么？舞台上立起林林总总的道具，彩绘披帛间看不清当年倚门回首的那个梦，只听到遥远的观众席上起伏的喝彩或骂声，曲中谢幕时漫天散落的金币砸碎脑际里青涩少年的背影，生活变成一座迷宫，看不到出口，望不见尽头，而命运的大手还在无情地搬弄迷宫里的砖石——在你看到出口的时候立起一座墙，在你委顿不前的时候拆出一扇窗，最后只能坐倒在地痛哭流涕，然后再擦干眼泪默默爬起。

大概生活就是这样的轮回吧。命运的恶趣味。

拉马克主义坚信着环境终究会改变生命，并且让这种扭曲过的性状一代代传承下去。尽管生物学和社会学现在还算是两个截然不同的分支，但是这种解释性的哲言偶尔也会说明一些问题。恶趣味的命运造就了恶趣味的人，包括坐在屏幕前的你。当围观成为习惯，当调戏成为本能，这时面对一些必须服从在你的力量之下的弱小者，你也会不自禁地去戏弄他。

### 题目描述

普通得甚至有些二逼的矮穷挫少年——Dios，不可避免地遇到了他生命中的劫数，白富美少女 Nyution。但是按照正常的校园故事的发展，Nyution 是无论如何不会喜欢上各方面条件都差到不行的 Dios 的。不过，Dios 还是面对 Nyution 颤抖着说出了那三个字。Nyution 既不想过分地让 Dios 伤心，又不想接受她根本看不上的 Dios，于是决定让 Dios 走一个建在她家后院里的迷宫花园——如果 Dios 能很快地从起点走到终点，证明他的聪明才智，Nyution 就答应他的表白。

当然 Nyution 敢这么说肯定是有准备的。Nyution 的花园可以看做一个迷宫，在迷宫内部有起点和终点。Dios 要从起点走到终点，并且他只能选择前后左右四个方向行走，而且显然不能走到篱笆上，也不能走出迷宫的边界。Nyution 经过仔细的调查发现，Dios 移动到相邻格子的耗时肯定是 1。同时，Nyution 将在 Dios 的挑战开始前，通过进行适当的路面调整，使 Dios 在南北方向（数据中的上下方向）的移动时间由 1 变成实数  $v$ 。首先，Nyution 不能让 Dios 过快地到达终点，这样她就得接受表白；其次，Nyution 也不想让 Dios 开了小宇宙之后还是过慢地到达终点，这样显得她在刁难 Dios。最后她确定了一个实数  $L$ ——就是最坏情况（也就是 Dios 最神勇威武耗时最短的情况）下，Dios 将花费  $L$  的时间由起点到

# Violet I

达终点。但是 Nyution 显然不会求此时的  $v$  值，于是她找到了一向以算法达人著称的你。你当然不会拒绝白富美 Nyution 的请求，决定帮她算出此时的  $v$ 。

由于 Nyution 不仅是白富美同时也是三好学生，所以她肯定不会给你一个无解的任务。并且，Nyution 的迷宫中一定没有水平的从起点到终点的通路。

## 输入格式

输入文件包含多个测试点。第一行包含一个整数，表示测试点的数目。

每个测试点的第一行包含实数  $L$  和两个整数  $R, C$ 。 $L$  的含义如上， $R$  表示 Nyution 的花园南北方向的长度， $C$  表示花园东西方向的长度。

之后  $R$  行为花园的描述，每行包含  $C$  个字符。其中空格（ASCII 码为 32）代表空地，S 代表起点，E 代表终点，# 代表篱笆。显然，起点和终点都是空地。

## 输出格式

对于每组测试数据，在单独的一行内输出  $v$  的值，保留 5 位小数。

## 样例输入

```
2
2.5 4 5
#####
#S   #
#   E#
#####
21 13 12
#####
#S##      #E#
#  ##  #  #  #
#    #  #  #  #
#### #  #  #  #
#    #  #  #  #
#  ## #  #  #
##   #  #  #  #
#### #  #  #  #
##   #  #  #  #
#  ## #      #
#      #      #
#####
```

## 样例输出

```
0.50000
0.21053
```

# Violet I

---

## 数据范围与约定

对于 20% 的数据，满足  $1 \leq R, C \leq 10$ 。

对于另外 20% 的数据，保证答案  $v$  的小数部分为 0，即  $\lfloor v \rfloor = v$ 。

对于 100% 的数据，满足  $1 \leq R, C \leq 100$ ，保证  $0 \leq v < 10$ 。

# Violet I

## 木偶

(puppet.pas/c/cpp)

### 题目描述

Nimo 是小镇上最出色的木偶剧演员。他经常带着自己的木偶们在附近的镇上演出。

为了方便起见，我们给每个木偶设定一个特征值  $p_i$ ，每个木偶都有一个原装的提线，

提线的特征值和木偶是相等的，即也为  $p_i$ 。

当一个木偶和一个提线的特征值相差超过 1 时，提线将不能组装到木偶上。

现在木偶和提线被 Nimo 不小心全部打乱了，Nimo 决定对这些木偶和提线重新配对。

他采取这样的策略：每次拿出一个木偶，再在未配对的提线里面随机抽一个可以组装到这个木偶上的提线，将它们配对，如果当前木偶没有可以配对的提线，Nimo 将扔掉这个木偶。

现在他想知道的是，在最坏情况下，他最多会扔掉多少个木偶。

### 输入格式

测试文件中包含若干个测试点。

每个测试点包含两行。第一行包含一个整数  $N$ ，表示有  $N$  个木偶。

接下来一行  $N$  个整数，第  $i$  个整数  $p_i$ ，表示第  $i$  个木偶及其原装提线的特征值。

### 输出格式

一行一个整数，表示最多可能扔掉多少个木偶。

### 样例输入

```
3
1 2 3
8
1 2 3 3 4 2 5 4
```

### 样例输出

```
1
2
```

### 样例说明

对于样例 1，Nimo 可能会把第 2 个木偶和第 1 个木偶的原配提线配对，第 3 个木偶和

# Violet I

---

第 2 个木偶的原配提线配对，这样第 3 个木偶就会被扔掉。

## 数据范围与约定

对于 10% 的数据，满足  $1 \leq N \leq 10$ 。

对于 20% 的数据，满足  $1 \leq N \leq 20$ 。

对于 100% 的数据，满足  $1 \leq N \leq 50$ ， $1 \leq p_i \leq 100$ 。

# Violet I

## 追风者

(chaser.pas/c/cpp)

### 题目描述

Jonathan 教授是一位著名的加拿大物理学家和气象学家。认识他的人都称他为“追风者”。除了他关于龙卷风的卓越研究成果在学术界有着举足轻重的影响力之外，还因为他经常以极其冒险的方法收集龙卷风的真实数据。实际上，他已经带领他的团队利用一辆装备有先进仪器的汽车，追逐龙卷风数百次了。

在夏季，Jonathan 教授生活的地区经常发生龙卷风。经过数年的研究，追风者发现了龙卷风构成和运动的很多规律。在卫星图片上，龙卷风显示为一个有着数米至数千米半径的圆。它的圆心会在两个位置之间进行直线往复运动，而且它会以相同的速率前进或者后退。在观察过一个龙卷风的移动之后，追风者会选择一条直的高速公路，沿着公路以汽车的最高速追逐这个龙卷风。

在整个过程中，追风者和龙卷风中心之间的最小距离被称为“观察距离”。观察距离对于研究活动有着至关重要的影响。如果距离太小，追风者就会有生命危险；如果距离太大的话，追风者又不能很好地观察龙卷风。在经历过数次危险和擦肩而过之后，追风者向你——他最优秀的学生之一求助。他只想知道一件事，那就是下一次的追逐，到底是会遇到危险，还是与龙卷风擦肩而过，抑或是一次完美的观察活动。

### 输入格式

输入文件包含若干个测试点。每个测试点包含三行，以如下的形式给出：

$$x_{w1} \ y_{w1} \ x_{w2} \ y_{w2} \ v_w$$
$$x_{t1} \ y_{t1} \ x_{t2} \ y_{t2} \ v_t$$
$$d_l \ d_u$$

在第一行， $(x_{w1}, y_{w1})$  表示追风者的起始位置， $(x_{w2}, y_{w2})$  是追风者选择的高速公路上的另一点，追风者会在途中经过该点。 $v_w$  是汽车的速率。你可以认为追风者会沿着那条无穷无尽的高速公路走到世界的尽头。

在第二行， $(x_{t1}, y_{t1})$  是龙卷风的起始点， $(x_{t2}, y_{t2})$  是龙卷风的转折点， $v_t$  是龙卷风的速率。换句话说，龙卷风会以不变的速率  $v_t$  在  $(x_{t1}, y_{t1})$  和  $(x_{t2}, y_{t2})$  之间做直线往复运动。

第三行说明观察距离如果小于  $d_l$ ，追风者就会遇到危险；如果观察距离大于  $d_u$ ，追风

# Violet I

---

者就会与龙卷风擦肩而过；否则就会是一次完美的观察活动。

你需要注意的是，追风者和龙卷风会恰好同时开始运动。

## 输出格式

对于每个测试点，在单独的一行内输出 `Dangerous`、`Perfect`、`Miss` 中的一个单词表示结果。

## 样例输入

```
0 0 1 0 2
10 -5 12 7 4
1.3 2.7
0 0 1 0 2
10 -5 12 7 1
0.3 0.4
```

## 样例输出

```
Dangerous
Perfect
```

## 数据范围与约定

对于全部测试点，所有的输入数据都是实数，并且

$$-2 \times 10^9 \leq x_{w1}, y_{w1}, x_{w2}, y_{w2}, x_{t1}, y_{t1}, x_{t2}, y_{t2} \leq 2 \times 10^9$$

$$1 \leq v_w, v_t \leq 2 \times 10^4$$

$$0 \leq d_l < d_u \leq 2 \times 10^6$$

保证观察距离与  $d_l$  或  $d_u$  的值相差都在  $10^{-5}$  以上。

对于 20% 的输入文件，测试点的数目为 1。

对于 60% 的输入文件，测试点的数目不大于 100。

对于 100% 的输入文件，测试点的数目不大于 5000。