# 全国青少年信息学奥林匹克
# 山东省省队选拔赛 2011 第一轮

## 第二试

**竞赛时间：2011 年 3 月 27 日**

| 题目名称 | 拦截导弹 | 工作安排 | 迷宫探险 |
|---|---|---|---|
| 目录 | missile | job | maze |
| 可执行文件名 | missile.exe | job.exe | maze.exe |
| 输入文件名 | missile.in | job.in | maze.in |
| 输出文件名 | missile.out | job.out | maze.out |
| 每个测试点时限 | 3s | 1s | 1s |
| 测试点数目 | 20 | 20 | 10 |
| 每个测试点分值 | 5 | 5 | 10 |
| 内存限制 | 512MB | 512MB | 512MB |
| 是否有部分分 | 有 | 无 | 无 |
| 题目类型 | 传统型 | 传统型 | 传统型 |

提交源程序须加后缀

| | | | |
|---|---|---|---|
| 对于 Pascal 语言 | pas | pas | pas |
| 对于 C 语言 | c | c | c |
| 对于 C++ 语言 | cpp | cpp | cpp |

**注意：最终测试时，所有编译命令均不打开任何优化开关。**

# 拦截导弹

## 【问题描述】

某国为了防御敌国的导弹袭击，发展出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度、并且能够拦截任意速度的导弹，但是以后每一发炮弹都不能高于前一发的高度，其拦截的导弹的飞行速度也不能大于前一发。某天，雷达捕捉到敌国的导弹来袭。由于该系统还在试用阶段，所以只有一套系统，因此有可能不能拦截所有的导弹。

在不能拦截所有的导弹的情况下，我们当然要选择使国家损失最小、也就是拦截导弹的数量最多的方案。但是拦截导弹数量的最多的方案有可能有多个，如果有多个最优方案，那么我们会随机选取一个作为最终的拦截导弹行动蓝图。

我方间谍已经获取了所有敌军导弹的高度和速度，你的任务是计算出在执行上述决策时，每枚导弹被拦截掉的概率。

## 【输入格式】

第一行包含一个正整数$n$，表示敌军导弹数量；

下面$n$行按顺序给出了敌军所有导弹信息：

第$i+1$行包含$2$个正整数$h_i$和$v_i$，分别表示第$i$枚导弹的高度和速度。

## 【输出格式】

输出包含两行。

第一行为一个正整数，表示最多能拦截掉的导弹数量；

第二行包含$n$个 0 到 1 之间的实数，第$i$个数字表示第$i$枚导弹被拦截掉的概率（你可以保留任意多位有效数字）。

## 【样例输入】

```
4
3 30
4 40
6 60
3 30
```

## 【样例输出】

```
2
0.33333 0.33333 0.33333 1.00000
```

## 【数据规模和约定】

对于 30%的数据，$1 \leq n \leq 1000$；

对于 100%的数据，$1 \le n \le 5 * 10^4$， $1 \le h_i, v_i \le 10^9$；
均匀分布着约 30%的数据，所有$v_i$均相等。
均匀分布着约 50%的数据，满足$1 \le h_i, v_i \le 1000$。

## 【评分标准】

对于每个测试点，若输出文件的第一行与标准输出相同，则得到该测试点 40%的分数，若输出文件的第二行的每个数与标准输出的误差不大于$10^{-4}$，则得到该测试点 60%的分数，两项可累加。

本题使用自定义校验器，为防止自定义校验器出错，即使你无法正确得出某一问的答案，也应在相应的位置随便输出一个数字。

# 工作安排

## 【问题描述】

你的公司接到了一批订单。订单要求你的公司提供$n$类产品，产品被编号为$1 \sim n$，其中第$i$类产品共需要$C_i$件。公司共有$m$名员工，员工被编号为$1 \sim m$，不同的员工能够制造的产品种类有所区别。一件产品必须完整地由一名员工制造，不可以由某名员工制造一部分配件后，再转交给另外一名员工继续进行制造。

我们用一个由 0 和 1 组成的$m * n$的矩阵$A$来描述每名员工能够制造哪些产品。矩阵的行和列分别被编号为$1 \sim m$和$1 \sim n$，$A_{i,j}$为 1 表示员工$i$能够制造产品$j$，为 0 表示员工$i$不能制造产品$j$。

如果公司分配了过多工作给一名员工，这名员工会变得不高兴。我们用愤怒值来描述某名员工的心情状态。愤怒值越高，表示这名员工心情越不爽，愤怒值越低，表示这名员工心情越愉快。员工的愤怒值与他被安排制造的产品数量存在某函数关系，鉴于员工们的承受能力不同，不同员工之间的函数关系也是有所区别的。

对于员工$i$，他的愤怒值与产品数量之间的函数是一个$S_i + 1$段的分段函数。当他制造第$1 \sim T_{i,1}$件产品时，每件产品会使他的愤怒值增加$W_{i,1}$，当他制造第$T_{i,1} + 1 \sim T_{i,2}$件产品时，每件产品会使他的愤怒值增加$W_{i,2}$……为描述方便，设$T_{i,0} = 0$，$T_{i,S_i+1} = +\infty$，那么当他制造第$T_{i,j-1} + 1 \sim T_{i,j}$件产品时，每件产品会使他的愤怒值增加$W_{i,j}$，$1 \le j \le S_i + 1$。

你的任务是制定出一个产品的分配方案，使得订单条件被满足，并且所有员工的愤怒值之和最小。由于我们并不想使用 Special Judge，也为了使选手有更多的时间研究其他两道题目，你只需要输出最小的愤怒值之和就可以了。

## 【输入格式】

第一行包含两个正整数$m$和$n$，分别表示员工数量和产品的种类数；

第二行包含$n$个正整数，第$i$个正整数为$C_i$；

以下$m$行每行$n$个整数描述矩阵$A$；

下面$m$个部分，第$i$部分描述员工$i$的愤怒值与产品数量的函数关系。每一部分由三行组成：第一行为一个非负整数$S_i$，第二行包含$S_i$个正整数，其中第$j$个正整数为$T_{i,j}$，如果$S_i = 0$那么输入将不会留空行（即这一部分只由两行组成）。第三行包含$S_i + 1$个正整数，其中第$j$个正整数为$W_{i,j}$。

**【输出格式】**

仅输出一个整数，表示最小的愤怒值之和。

**【样例输入】**

```
2 3
2 2 2
1 1 0
0 0 1
1
2
1 10
1
2
1 6
```

**【样例输出】**

```
24
```

**【数据规模和约定】**

存在 30%的数据，$m,n \le 30$；
均匀分布着约 30%的数据，满足$S_i = 0$；
均匀分布着约 30%的数据，满足$S_i \le 1$（不包含上述$S_i = 0$的数据）；

对于 100%的数据，$1 \le m,n \le 250$，$0 \le S_i \le 5$，$0 \le A_{i,j} \le 1$，$0 < T_{i,j} < T_{i,j+1}$，$0 < W_{i,j} < W_{i,j+1}$，所有数据不大于$10^5$。

# 迷宫探险

## 【问题描述】

这是一个单人游戏。

游戏开始时，玩家控制的人物出生在迷宫的某个位置，玩家的目标是控制人物走到迷宫的某个出口（出口可能有多个）。迷宫里有$K$类陷阱（用“A”、“B”、“C”……表示，相同字母代表相同类型的陷阱），每类陷阱可能是有害的或无害的，而在游戏开始时玩家并不知道哪些陷阱是有害的，哪些是无害的。同一类陷阱的状态相同，即用同一个字母标志的陷阱要么全部有害，要么全部无害，不会发生一部分有害而另一部分无害的情况。任何陷阱状态的组合都有一个发生概率，考虑下例：

当$K = 2$时，迷宫内共有两类陷阱，分别用“A”和“B”表示，陷阱状态的组合共有 4 种：

1、“A”是无害陷阱，“B”是无害陷阱。

2、“A”是有害陷阱，“B”是无害陷阱；

3、“A”是无害陷阱，“B”是有害陷阱；

4、“A”是有害陷阱，“B”是有害陷阱；

下面给出了一个合法的概率表格：

| | “A”是无害陷阱 | “A”是有害陷阱 |
|---|---|---|
| “B”是无害陷阱 | 36% | 24% |
| “B”是有害陷阱 | 24% | 16% |

当$K = 3$时，会有 8 种不同的陷阱状态组合，如果我们依然坚持使用概率表格，那么这个表格将会是三维的（$2 * 2 * 2$，每一维对应着一类陷阱）。当$K \geq 3$时，这将使得题目难以描述。因此我们使用一个大小为$2^K$的数组$p$来描述每种情况发生的可能性，$p$的下标范围为$0 \sim 2^K - 1$。

$p$是这样生成的：

对于每个可能的陷阱状态组合，考虑所有$K$类陷阱，令 1 表示某个陷阱有害，0 表示某个陷阱无害，把“A”作为二进制数的第 0 位（从右边开始计数），“B”作为第 1 位，“C”作为第 2 位……通过以上操作，我们可以得到一个 $K$ 位的二进制数，把它转化成十进制后，$2^K$种陷阱状态的组合将会与整数$0 \sim 2^K - 1$一一对应。

定义$s$表示$p$中所有元素和，即：

$$s = \sum_{i=0}^{2^K-1} p_i$$

则陷阱状态组合$i$出现的概率为$p_i / s$。上述表格对应的一个合法数组$p$是：

$$p_0 = 36$$
$$p_1 = 24$$
$$p_2 = 24$$
$$p_3 = 16$$

当然同一个概率表格可能会对应多个数组$p$（事实上有无数个数组$p$能够迎合表格数据），例如上述表格同时也对应着下面的数组$p$：

$$p_0 = 72$$
$$p_1 = 48$$
$$p_2 = 48$$
$$p_3 = 32$$

玩家控制的人物初始情况下有$H$点生命，当人物踏上某个陷阱时，如果这个陷阱是有害的，那么会损失 1 点生命，否则这个陷阱是无害的，不损失生命。无论上述哪种情况发生，玩家会立刻得到这个陷阱的信息（有害或无害）。一旦生命小于等于 0，玩家控制的人物会立刻死亡。

迷宫可以看作$m * n$的方格地图，每个元素可能是：

“.”：表示这是平地，可以通过；

“#”：表示这是墙，不能通过；

“A”，“B”，“C”……：表示这是一个陷阱；

“$”：表示这是起点，地图中有且仅有一个；

“@”：表示这是终点，地图中可以有多个，也可以一个也没有。

人物可以向上下左右四个方向行走，不可以走对角线，也不可以走出地图。

给定$m * n$的地图、$K$、$H$以及大小为$2^K$的概率数组。你的任务是求出在执行最优策略时，人物能活着走出迷宫的概率。

## 【输入格式】

第一行包含 4 个整数，分别表示$m$、$n$、$K$、$H$；

下面$m$行每行$n$个字符描述迷宫地图；

最后一行包含$2^K$个非负整数描述数组$p$，数组下标从 0 开始。

## 【输出格式】

仅包含一个数字，表示在执行最优策略时，人物活着走出迷宫的概率。四舍五入保留 3 位小数。

## 【样例输入 1】

```
4 3 2 1
.$.
A#B
A#B
.@.
30 30 20 20
```

## 【样例输出 1】

```
0.600
```

## 【样例说明 1】

向右边走，经过“B”，“B”为有害陷阱的概率为$(20 + 20)/(30 + 30 + 20 + 20) = 0.4$，若“B”为有害陷阱那么人物就死掉了，游戏失败，否则玩家得知“B”

是无害陷阱，继续经过另一个“B”达到终点，胜利的概率为 0.6。

**【样例输入 2】**

```
4 3 2 2
.$.
A#B
A#B
.@.
30 30 20 20
```

**【样例输出 2】**

```
0.800
```

**【样例说明 2】**

向左边走，经过“A”，“A”为有害陷阱的概率为$(30+20)/(30+30+20+20)=0.5$。若“A”为有害陷阱，那么损失一点生命，转到右边尝试“B”，要想成功到达终点，此时“B”必须为无害陷阱，而在“A”是有害陷阱的前提下，“B”是无害陷阱的概率是$30/(30+20)=0.6$，故这种情况发生的概率为$0.5*0.6=0.3$。若“A”是无害陷阱，玩家可以控制人物连续通过两个“A”到达终点，这种情况发生的概率$0.5$。所以答案为$0.3+0.5=0.8$。

**【样例输入 3】**

```
4 3 2 3
.$.
A#B
A#B
.@.
30 30 20 20
```

**【样例输出 3】**

```
1.000
```

**【样例说明 3】**

玩家控制的人物有 3 点生命，但最多只需要经过两个陷阱，所以任意选左路或右路走过去就可以到达终点了。

**【样例输入 4】**

```
4 3 3 2
.$.
A#B
```

```
A#C
@@@
143 37 335 85 95 25 223 57
```

## 【样例输出 4】

```
0.858
```

## 【数据规模和约定】

| 测试点编号 | $m$ | $n$ | $K$ | $H$ |
|---|---|---|---|---|
| 1 | 29 | 28 | 5 | 1 |
| 2 | 28 | 20 | 4 | 1 |
| 3 | 25 | 30 | 1 | 1 |
| 4 | 25 | 30 | 1 | 2 |
| 5 | 25 | 30 | 1 | 3 |
| 6 | 5 | 5 | 4 | 4 |
| 7 | 12 | 11 | 4 | 5 |
| 8 | 19 | 17 | 5 | 3 |
| 9 | 23 | 25 | 5 | 4 |
| 10 | 30 | 29 | 5 | 5 |

对于 100%的数据，$0 \le p_i \le 10^5$，且至少有一个$p_i > 0$。