“AMD”杯

第二十五届全国信息学奥林匹克竞赛

NOI 2008

第二试

竞赛时间：2008 年 7 月 31 日上午 8:00-13:00

题目名称	奥运物流	糖果雨	赛程安排
目录	trans	candy	match
可执行文件名	trans	candy	match
输入文件名	trans.in	candy.in	match1.in~match10.in
输出文件名	trans.out	candy.out	match1.out~match10.out
每个测试点时限	1s	2s	N/A
内存限制	128M	128M	N/A
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	有
题目类型	传统	传统	提交答案

提交源程序须加后缀

对于 Pascal 语言	trans.pas	candy.pas	N/A
对于 C 语言	trans.c	candy.c	N/A
对于 C++ 语言	trans.cpp	candy.cpp	N/A

注意：最终测试时，所有编译命令均不打开任何优化开关

奥运物流

【问题描述】

2008 北京奥运会即将开幕，举国上下都在为这一盛事做好准备。为了高效率、成功地举办奥运会，对物流系统进行规划是必不可少的。

物流系统由若干物流基站组成，以 1…N 进行编号。每个物流基站 i 都有且仅有一个后继基站 S_i，而可以有多个前驱基站。基站 i 中需要继续运输的物资都将被运往后继基站 S_i，显然一个物流基站的后继基站不能是其本身。编号为 1 的物流基站称为控制基站，从任何物流基站都可将物资运往控制基站。注意控制基站也有后继基站，以便在需要时进行物资的流通。在物流系统中，高可靠性与低成本是主要设计目。对于基站 i，我们定义其"可靠性" $R(i)$ 如下：

设物流基站 i 有 w 个前驱基站 $P_1, P_2, \cdots P_w$，即这些基站以 i 为后继基站，则基站 i 的可靠性 $R(i)$满足下式：

$$R(i) = C_i + k\sum_{j=1}^{w} R(P_j)$$

其中 C_i 和 k 都是常实数且恒为正，且有 k 小于 1。

整个系统的可靠性与控制基站的可靠性正相关，我们的目标是通过修改物流系统，即更改某些基站的后继基站，使得控制基站的可靠性 $R(1)$尽量大。但由于经费限制，最多只能修改 m 个基站的后继基站，并且，控制基站的后继基站不可被修改。因而我们所面临的问题就是，如何修改不超过 m 个基站的后继，使得控制基站的可靠性 $R(1)$最大化。

【输入格式】

输入文件 trans.in 第一行包含两个整数与一个实数，N, m, k。其中 N 表示基站数目，m 表示最多可修改的后继基站数目，k 分别为可靠性定义中的常数。

第二行包含 N 个整数，分别是 $S_1, S_2 \ldots S_N$，即每一个基站的后继基站编号。

第三行包含 N 个正实数，分别是 $C_1, C_2 \ldots C_N$，为可靠性定义中的常数。

【输出格式】

输出文件 trans.out 仅包含一个实数，为可得到的最大 $R(1)$。精确到小数点两位。

【输入样例】

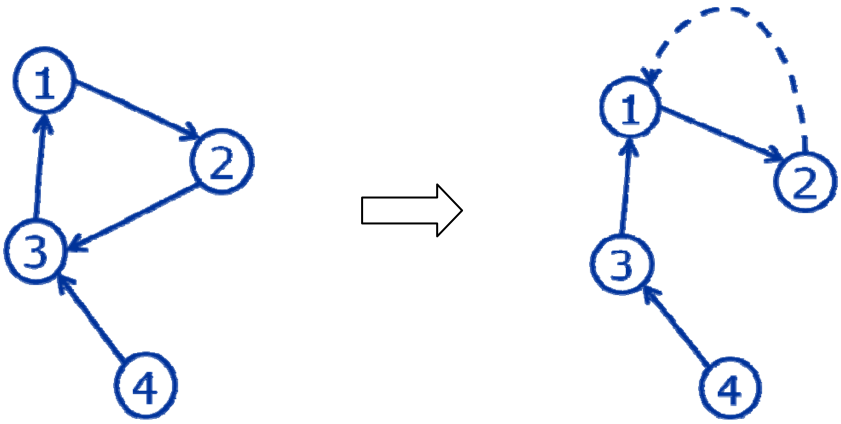
4 1 0.5
2 3 1 3
10.0 10.0 10.0 10.0

【输出样例】

30.00

【样例说明】

原有物流系统如左图所示，4 个物流基站的可靠性依次为 22.8571，21.4286，25.7143，10。



最优方案为将 2 号基站的后继基站改为 1 号，如右图所示。 此时 4 个基站的可靠性依次为 30，25，15，10。

【数据规模和约定】

本题的数据，具有如下分布：

测试数据编号	N	M
1	≤ 6	≤ 6
2	≤ 12	≤ 12
3	≤ 60	0
4	≤ 60	1
5	≤ 60	N-2
6~10	≤ 60	≤ 60

对于所有的数据，满足 $m \le N \le 60$，$C_i \le 10^6$，$0.3 \le k < 1$，请使用双精度实数，无需考虑由此带来的误差。

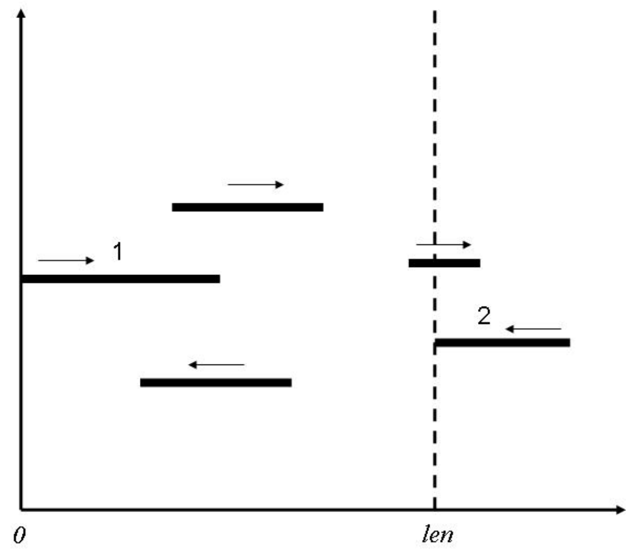
糖果雨

【问题描述】

有一个美丽的童话：在天空的尽头有一个"糖果国"，这里大到摩天大厦，小到小花小草都是用糖果建造而成的。更加神奇的是，天空中飘满了五颜六色的糖果云，很快糖果雨密密麻麻从天而落，红色的是草莓糖，黄色的是柠檬糖，绿色的是薄荷糖，黑色的是巧克力糖……这时糖果国的小朋友们便会拿出大大小小的口袋来接天空中落下的糖果，拿回去与朋友们一起分享。

对糖果情有独钟的小 Z 憧憬着能够来到这样一个童话的国度。所谓日有所思，夜有所梦，这天晚上小 Z 梦见自己来到了"糖果国"。他惊喜地发现，任何时候天空中所有的云朵颜色都不相同，不同颜色的云朵在不断地落下相应颜色的糖果。更加有趣的是所有的云朵都在做着匀速往返运动，不妨想象天空是有边界的，而所有的云朵恰好在两个边界之间做着往返运动。每一个单位时间云朵向左或向右运动一个单位，<u>当云朵的左界碰到天空的左界，它会改变方向向右运动</u>；<u>当云朵完全移出了天空的右界，它会改变方向向左运动。</u>

我们不妨把天空想象为一个平面直角坐标系，而云朵则抽象为线段(线段可能退化为点)：



如上图，不妨设天空的左界为 0，右界为 *len*。图中共有 5 片云朵，其中标号为 1 的云朵恰好改变方向向右运动，标号为 2 的云朵恰好改变方向向左运动。忽略云朵的纵坐标，它们在运动过程中不会相互影响。

小 Z 发现天空中会不断出现一些云朵(某个时刻从某个初始位置开始朝某个方向运动)，而有的云朵运动到一定时刻就会从天空中消失，而在运动的过程中糖果在不断地下落。小 Z 决定拿很多口袋来接糖果，口袋容量是无限的，但袋口大小却是有限的。例如在时刻 T 小 Z 拿一个横坐标范围为$[L, R]$的口袋来接糖果，如果$[L, R]$存在一个位置 x，该位置有某种颜色的糖果落下，则认为该口袋可接到此种颜色的糖果。极端情况下，袋口区间可能是一个点，譬如[0,0]、[1,1]，

但仍然可以接到相应位置的糖果。通常可以接到的糖果总数会很大，因而小 Z 想知道每一次(即拿出口袋的一瞬间)他的口袋可以接到多少种不同颜色的糖果。糖果下落的时间忽略不计。

【输入格式】

输入文件 candy.in 的第一行有两个正整数 n, len，分别表示事件总数以及天空的“边界”。

接下来 n 行每行描述一个事件，所有的事件按照输入顺序依次发生。每行的第一个数 k（$k = 1$，2，3）分别表示事件的类型，分别对应三种事件：**插入事件**，**询问事件**以及**删除事件**。输入格式如下：

事件类型	输入格式	说明
插入事件 (天空中出现了一片云朵)	1 T_i C_i L_i R_i D_i	时刻 T_i，天空中出现了一片坐标范围为[L_i, R_i]，颜色为 C_i 的云朵，初始的时候云朵运动方向为向左(D_i = -1)或向右(D_i = 1)。满足 $0 \le L_i \le R_i \le len$，$D_i$ = -1 或 1。 数据保证任何时刻空中不会出现两片颜色相同的云朵。
询问事件 (询问一个口袋可以接到多少种不同颜色的糖果)	2 T_i L_i R_i	时刻 T_i，小 Z 用一个坐标范围为[L_i, R_i]的大口袋去接糖果，询问可以接到多少种不同的糖果。满足 $0 \le L_i \le R_i \le len$。
删除事件 (天空中一片云朵消失了)	3 T_i C_i	时刻 T_i，颜色为 C_i 的云朵从天空消失中。数据保证当前天空中一定存在一片颜色为 C_i 的云朵。

【输出格式】

对于每一个询问事件，输出文件 candy.out 中应包含相应的一行，为该次询问的答案，即口袋可以接到多少种不同的糖果。

【输入样例】

```
10 10
1 0 10 1 3 -1
2 1 0 0
2 11 0 10
2 11 0 9
```

1 11 13 4 7 1
2 13 9 9
2 13 10 10
3 100 13
3 1999999999 10
1 2000000000 10 0 1 1

【输出样例】

1
1
0
2
1

【样例说明】

共 10 个事件，包括 3 个插入事件，5 个询问事件以及 2 个删除事件。

时刻 0，天空中出现一片颜色为 10 的云朵，初始位置为[1, 3]，方向向左。

时刻 1，范围为[0, 0]的口袋可以接到颜色为 10 的糖果(云朵位置为[0, 2])。

时刻 11，范围为[0,10]的口袋可以接到颜色为 10 的糖果(云朵位置为[10, 12])。

时刻 11，范围为[0, 9]的口袋不能接到颜色为 10 的糖果(云朵位置为[10, 12])。

时刻 11, 天空中出现一片颜色为 13 的云朵, 初始位置为[4, 7], 方向向右。

时刻 13，范围为[9, 9]的口袋可以接到颜色为 10(云朵的位置为[8, 10])和颜色为 13(云朵的位置为[6, 9])两种不同的糖果。

时刻 13，范围为[10, 10]的口袋仅仅可以接到颜色为 10 的一种糖果(云朵的位置为[8, 10])，而不可以接到颜色为 13 的糖果(云朵的位置为[6, 9]),。

时刻 100, 颜色为 13 的云朵从天空中消失。

时刻 1999999999，颜色为 10 的云朵从天空中消失。

时刻 2000000000，天空中又出现一片颜色为 10 的云朵，初始位置为[0, 1]，方向向右。

【数据规模和约定】

对于所有的数据，$0 \le Ti \le 2000000000$，$1 \le C_i \le 1000000$。

数据保证$\{T_i\}$为非递减序列即 $T_1 \le T_2 \le \cdots \le T_{n-1} \le T_n$。

对于所有的**插入事件**，令 $P_i = R_i - L_i$，即 P_i 表示每片云朵的长度。

数据编号	n	len	P_i	数据编号	n	len	P_i
1	20	10	$\le len$	6	150000	1000	≤ 3
2	200	100	$\le len$	7	200000	1000	≤ 3
3	2000	1000	$\le len$	8	100000	1000	$\le len$
4	100000	10	$\le len$	9	150000	1000	$\le len$
5	100000	100	≤ 2	10	200000	1000	$\le len$

赛程安排

【问题描述】

随着奥运的来临，同学们对体育的热情日益高涨。在 NOI2008 来临之际，学校正在策划组织一场乒乓球赛。小 Z 作为一名狂热的乒乓球爱好者，这正是他大展身手的好机会，于是他摩拳擦掌，积极报名参赛。

本次乒乓球赛采取淘汰赛制，获胜者晋级。恰好有 n (n 是 2 的整数次幂，不妨设 $n = 2^k$)个同学报名参加，因此第一轮后就会有 2^{k-1} 个同学惨遭淘汰，另外 2^{k-1} 个同学晋级下一轮；第二轮后有 2^{k-2} 名同学晋级下一轮，… 依次类推，直到 k 轮后决出冠亚军：具体的，每个人都有一个 1~n 的**初始编号**，其中小 Z 编号为 1，所有同学的编号都不同，他们将被分配到 n 个**位置**中，然后按照类似下图的赛程进行比赛：

<table>
<tr><td>位置 1</td><td>位置 2</td><td>位置 3</td><td>位置 4</td><td>位置 5</td><td>位置 6</td><td>位置 7</td><td>位置 8</td></tr>
<tr><td colspan="2">位置 1 和位置 2 的胜者</td><td colspan="2">位置 3 和位置 4 的胜者</td><td colspan="2">位置 5 和位置 6 的胜者</td><td colspan="2">位置 7 和位置 8 的胜者</td></tr>
<tr><td colspan="4">位置 1234 中的胜者</td><td colspan="4">位置 5678 的胜者</td></tr>
<tr><td colspan="8">冠军</td></tr>
</table>

n=8 时比赛的赛程表

为了吸引更多的同学参加比赛，本次比赛的奖金非常丰厚。在第 i 轮被淘汰的选手将得到奖金 a_i 元，而冠军将获得最高奖金 a_{k+1} 元。显然奖金应满足 $a_1 < a_2 < \cdots < a_{k+1}$.

在正式比赛前的热身赛中，小 Z 连连败北。经过认真分析之后，他发现主要的失败原因不是他的球技问题，而是赢他的这几个同学在球风上刚好对他构成相克的关系，所以一经交手，他自然败阵。小 Z 思索：如果在正式比赛中能够避开这几位同学，该有多好啊!

假设已知选手两两之间交手的胜率，即选手 A 战胜选手 B 的概率为 $P_{A,B}$ (保证 $P_{A,B} + P_{B,A}$=1)。于是小 Z 希望能够通过确定比赛的对阵形势（重新给每个选手安排位置），从而能够使得他获得尽可能多的奖金。你能帮助小 Z 安排一个方案，使得他这场比赛期望获得的奖金最高么？

【输入格式】

这是一道提交答案型试题，所有的输入文件 match*.in 已在相应目录下。

输入文件 match*.in 第一行包含一个正整数 n，表示参赛的总人数，数据保证存在非负整数 k，满足 $2^k = n$。

接下来 n 行，每行有 n 个 0 到 1 间的实数 $P_{i,j}$，表示编号为 i 的选手战胜编号为 j 的选手的概率，每个实数精确到小数点后两位。特别注意 $P_{i,i} = 0.00$。

接下来 k+1 行，每行一个整数分别为晋级各轮不同的奖金，第 i 行的数为 a_i。

【输出格式】

输出文件 match*.out 包括 n 行，第 i 行的数表示位于第 i 个位置的同学的编号，**要求小 Z 的编号一定位于第 1 个位置**。

【输入样例】

```
4
0.00 0.70 0.60 0.80
0.30 0.00 0.60 0.40
0.40 0.40 0.00 0.70
0.20 0.60 0.30 0.00
1
2
3
```

【输出样例】

```
1
4
2
3
```

【样例说明】

第一轮比赛过后，编号为 1 的选手(小 Z)晋级的概率为 80%，编号为 2 的选手晋级的概率为 60%，编号为 3 的选手晋级的概率为 40%，编号为 4 的选手晋级的概率为 20%。

第二轮（决赛），编号为 1 的选手（小 Z）前两轮均获胜的概率为 80% * (60%*70% + 40%*60%) = 52.8%，因此, 小 Z 在第一轮失败的概率 P_1=1-0.8=0.2, 第一轮胜出但第二轮败北的概率 P_2=0.8-0.528=0.272, 获得冠军的概率 P_3=0.528。从而，期望奖金为 0.2*1 + (0.8-0.528)*2 + 0.528*3 = 2.328。

【如何测试你的输出】

我们提供 `match_check` 这个工具来测试你的输出文件是否可接受。 使用这个工具的测试方法是在终端中使用命令：

`./match_check` 测试数据编号

例如：`./match_check 10` 表示测试你的 match10.out 是否合法。

调用这个程序后，`match_check`将根据你得到的输出文件给出测试的结果，

其中包括：

非法退出：未知错误；

Format error：输出文件格式错误；

Not a permutation：输出文件不是一个1~n的排列；

OK.Your answer is xxx：输出文件可以被接受，xxx为对应的期望奖金。

【评分方法】

每个测试点单独评分。

对于每一个测试点，如果你的输出文件不合法，如文件格式错误、输出解不符合要求等，该测试点得0分。否则如果你的输出的期望奖金为 $your\_ans$，参考期望奖金为 our_ans，我们还设有一个用于评分的参数 d，你在该测试点中的得分如下：

如果 $your\_ans > our\_ans$，得12分。

如果 $your\_ans < our\_ans*d$，得1分。

否则得分为：

$$\left\lfloor \frac{your\_ans - our\_ans*d}{our\_ans - our\_ans*d}*8 \right\rfloor + 2$$

【提示】

“数学期望”

数学期望是随机变量最基本的数字特征之一。它反映随机变量平均取值的大小，又称期望或均值。它是简单算术平均的一种推广。例如某城市有10万个家庭，没有孩子的家庭有1000 个，有一个孩子的家庭有9万个，有两个孩子的家庭有6000个，有3个孩子的家庭有3000个，则该城市中任一个家庭中孩子的数目是一个随机变量，它可取值0，1，2，3，其中取0的概率为0.01，取1的概率为0.9，取2的概率为0.06，取3的概率为0.03，它的数学期望为0×0.01＋1×0.9＋2×0.06＋3×0.03等于1.11，即此城市一个家庭平均有小孩1.11个。

本题中期望值的计算：假设小Z在第一轮被打败的概率为 P_1，第一轮胜利且在第二轮被打败的概率为 P_2, 前两轮胜利且在第三轮被打败的概率为 P_3……，那么小Z的期望奖金为：

$$P_1 * a_1 + P_2 * a_2 + \ldots + P_{k+1} * a_{k+1}$$

【特别提示】

请妥善保存输入文件*.in 和你的输出*.out，及时备份，以免误删。