第 29 届全国青少年信息学奥林匹克竞赛

上海市选拔赛

SHTSC 2012

第一试

竞赛时间：2012 年 4 月 22 日上午 7:30-12:00

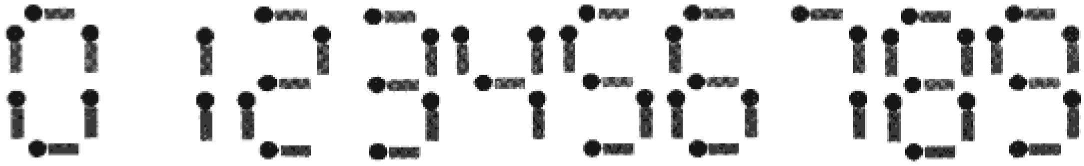
题目名称	火柴游戏	信用卡凸包	随机树
英文名称	match	card	random
源程序	match.pas 或 match.cpp	card.pas 或 card.cpp	random.pas 或 random.cpp
输入文件名	match.in	card.in	random.in
输出文件名	match.out	card.out	random.out
每个测试点时限	1 秒	1 秒	1 秒
内存限制	128M	128M	128M
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	否
题目类型	传统	传统	传统

上海市科技艺术教育中心

火柴游戏

【问题描述】

小明非常喜欢玩火柴游戏：首先用火柴棒摆出一个可能是错误的等式，然后通过添加、删除或移动火柴棒，使得等式成立。下图展示每个数字的样子：



我们只考虑形如“A = B”的式子，其中 A 和 B 是两个具有相同位数的整数。小明可进行三种操作：

1. 在任意位置添加一根火柴棒；
2. 从任意位置删除一根火柴棒；
3. 将任意一根火柴棒移动到另一个位置。

在完成所有操作后，等号两侧必须都是合法的数字，且完全相等。我们约定：

1. 小明不能消除任何数字，也就是说，可以删除一个数字的部分火柴，但不能令它消失；
2. 小明不能增加任何数字，也就是说，可以在一个已有的数字上添加火柴，或将火柴移动到一个已有的数字上，但不能凭空增加一个数字；
3. 小明不能拆分或者合并数字，比如将一个 8 变成两个 1，或者将两个 1 合并成一个 8；
4. 其中代表 1 的火柴棒必须靠右边摆放，放在左边不是有效的数字。

每种操作都有一定的代价：

- 对一个添加操作，如果这是第 i 次进行添加操作，这一步的费用为 $p_1*i + q_1$
- 对一个删除操作，如果这是第 i 次进行删除操作，这一步的费用为 $p_2*i + q_2$
- 对一个移动操作，如果这是第 i 次进行移动操作，这一步的费用为 $p_3*i + q_3$

例如，小明在游戏中添加了 3 根火柴，移动了 1 根火柴，删除了 2 根火柴，那么他总的花费为$[(p_1*1 + q_1) + (p_1*2 + q_1) + (p_1*3 + q_1)] + (p_3*1 + q_3) + [(p_2*1 + q_2) + (p_2*2 + q_2)]$。

小明想知道，他如何才能用最少的花费使等式成立。你能写个程序帮助他吗？

【输入格式】

第 1 行，一个整数 L，表示等式中两个数的位数。

第 2-3 行，每行各一个长度为 L、仅由数字构成的字符串，表示等式两侧的数。

第 4 行，给出六个不超过 100 的非负整数 $p_1, q_1, p_2, q_2, p_3, q_3$。

【输出格式】

输出一行，包含一个整数，为使等式成立的最小的操作代价。

【输入样例 1】

```
2
46
78
0 1 0 1 0 1
```

【输出样例 1】

```
2
```

【输入样例 2】

```
2
23
52
1 1 1 1 1 1
```

【输出样例 2】

```
2
```

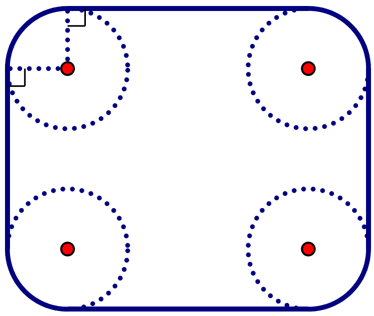
【数据规模】

对于 30%数据，有 $L \le 20$，且 $p_1 = p_2 = p_3 = 0$；
对于 60%数据，有 $L \le 100$；
对于 100%数据，有 $L \le 200$。

信用卡凸包

【问题描述】

信用卡是一个矩形，唯四个角作了圆滑处理，使它们都是与矩形的两边相切的 1/4 圆，如下图所示。现在平面上有一些规格相同的信用卡，试求其凸包的周长。注意凸包未必是多边形，因为它可能包含若干段圆弧。



【输入格式】

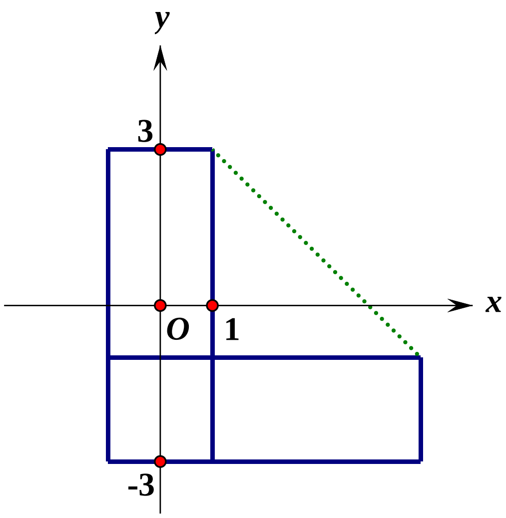
输入的第一行是一个正整数 n，表示信用卡的张数。

第二行包含三个实数 a, b, r，分别表示信用卡（圆滑处理前）竖直方向的长度、水平方向的长度，以及 1/4 圆的半径。

之后 n 行，每行包含三个实数 x, y, θ，分别表示一张信用卡中心（即对角线交点）的横、纵坐标，以及绕中心**逆时针**旋转的**弧度**。

【输出格式】

输出只有一行，包含一个实数，表示凸包的周长，四舍五入精确到小数点后 2 位。



【输入样例 1】

```
2
6.0 2.0 0.0
0.0 0.0 0.0
2.0 -2.0 1.5707963268
```

【输出样例 1】

```
21.66
```

【样例 1 说明】

本样例中的 2 张信用卡的轮廓在上图中用实线标出，如果视 1.5707963268

为$\frac{\pi}{2}$，则其凸包的周长为$16 + 4\sqrt{2}$。

【输入样例 2】

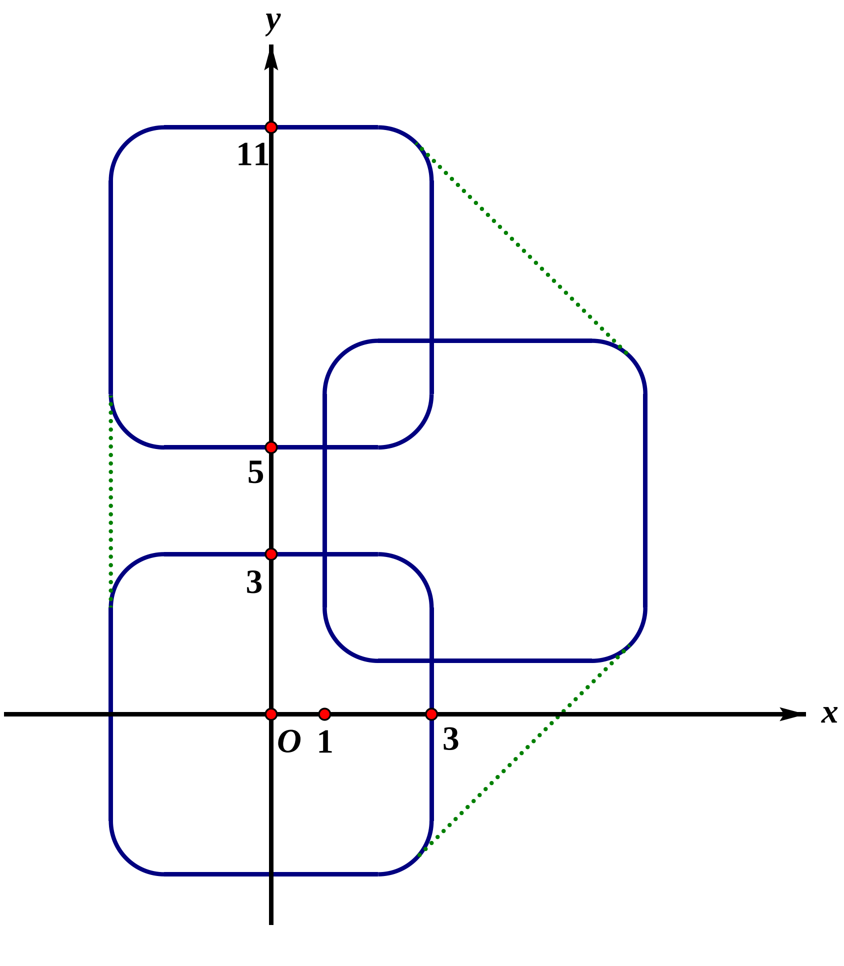
```
3
6.0 6.0 1.0
4.0 4.0 0.0
0.0 8.0 0.0
0.0 0.0 0.0
```

【输出样例 2】

```
41.60
```

【样例 2 说明】

本样例中的 3 张信用卡的轮廓在下图中用实线标出，其凸包的周长为。



【输入样例 3】

```
3
```

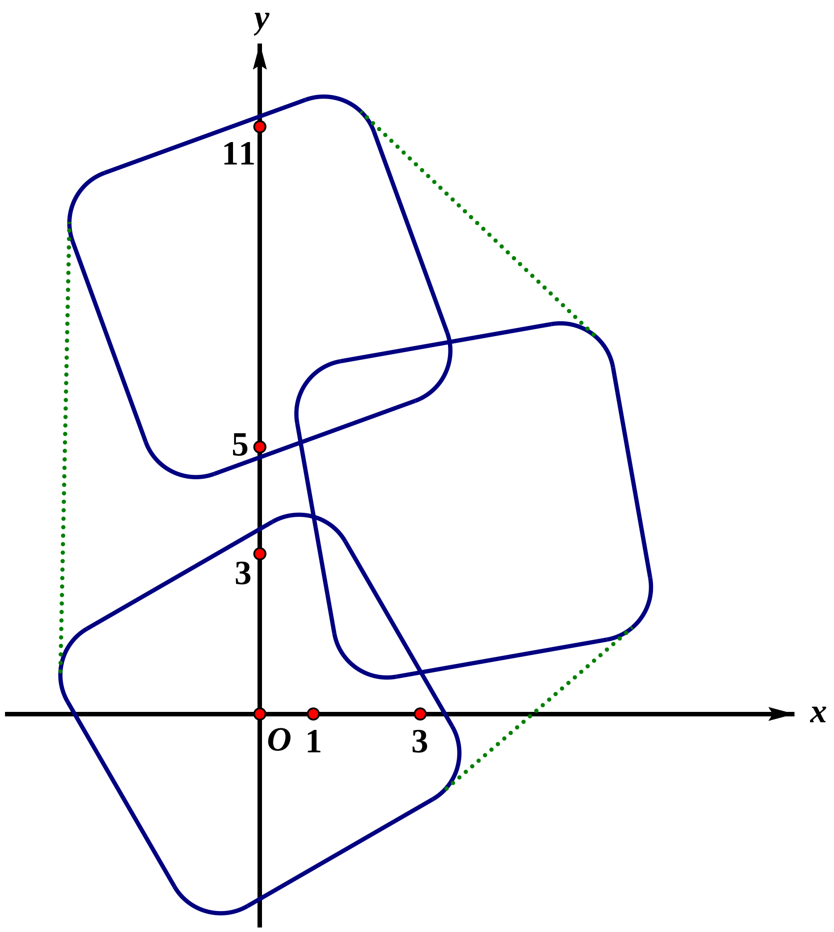
```
6.0 6.0 1.0
4.0 4.0 0.1745329252
0.0 8.0 0.3490658504
0.0 0.0 0.5235987756
```

【输出样例 3】

```
41.63
```

【样例 3 说明】

本样例中的 3 张信用卡的轮廓在下图中用实线标出，其凸包的周长约为 41.628267652。



【数据规模】

测试数据编号	n	r	θ
1	$n = 1$	/	/
2	$n = 2$	$r = 0.0$	所有的 θ 均为 0.0
3	$n = 2$	/	所有的 θ 均为 0.0
4	$n = 2$	$r = 0.0$	/
5	$n = 2$	/	/

6	$1 \le n \le 100$	/	所有的 θ 均为 0.0
7	$1 \le n \le 100$	/	/
8	$1 \le n \le 10{,}000$	/	所有的 θ 均为 0.0
9	$1 \le n \le 10{,}000$	$r = 0.0$	/
10	$1 \le n \le 10{,}000$	/	/

对于 100%的数据，有 $0.1 \le a, b \le 1000000.0$，以及 $0.0 \le r < \min\{a/4, b/4\}$，对所有的信用卡，有$|x|, |y| \le 1000000.0$，以及 $0 \le \theta < 2\pi$。

【提示】

本题可能需要使用数学库中的三角函数。不熟悉使用方法的选手，可以参考下面的程序及其输出结果：

Pascal 程序

```
uses math;
const Pi = 3.141592653589793;

begin
   writeln(sin(30.0 / 180.0 * Pi) : 0 : 10);
   writeln(cos(60.0 / 180.0 * Pi) : 0 : 10);
   writeln(tan(45.0 / 180.0 * Pi) : 0 : 10);
   writeln(arcsin(1.0) : 0 : 10);
   writeln(arccos(0.0) : 0 : 10);
   writeln(arctan(1.0) : 0 : 10);
end.
```

输出结果

```
0.5000000000
0.5000000000
1.0000000000
1.5707963268
1.5707963268
0.7853981634
```

C++程序

```
#include <iostream>
#include <math.h>
using namespace std;
const double Pi = 3.141592653589793;

int main()
{
   cout.setf(ios::fixed);
   cout.precision(10);
   cout<<sin(30.0 / 180.0 * Pi)<<endl;
   cout<<cos(60.0 / 180.0 * Pi)<<endl;
   cout<<tan(45.0 / 180.0 * Pi)<<endl;
   cout<<asin(1.0)<<endl;
   cout<<acos(0.0)<<endl;
   cout<<atan(1.0)<<endl;
   return 0;
}
```

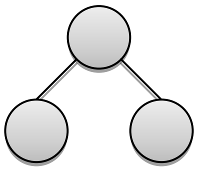
输出结果

```
0.5000000000
0.5000000000
1.0000000000
1.5707963268
1.5707963268
0.7853981634
```

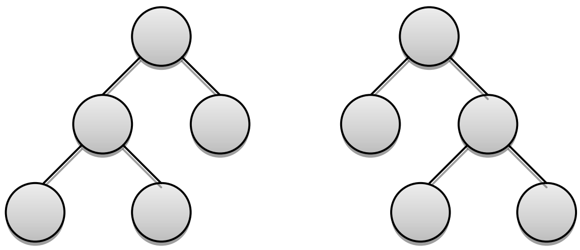
随机树

【问题描述】

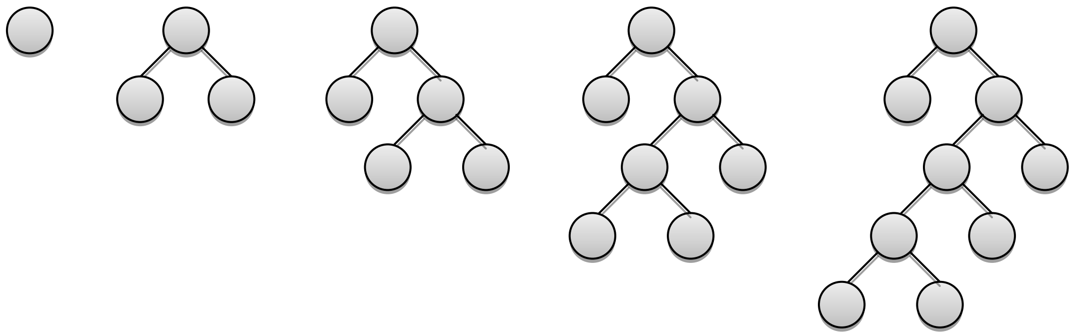
一棵含 n 个叶结点的二叉树可以通过如下方式生成。初始时只有根结点。首先，将根结点展开（本题中的“展开”是指给一个叶结点添上左、右两个子结点）：



然后，等概率地随机将两个叶结点中的一个展开，即生成以下两棵树之一：
之后，每次在当前二叉树的所有叶结点中，等概率地随机选择一个，将其展开。



不断地重复这一操作，直至产生 n 个叶结点为止。例如，某棵含 5 个叶结点的二叉树可能按如下步骤生成。



对于按这种方式随机生成的一棵含 n 个叶结点二叉树，求（1）**叶结点平均深度**的数学期望值；（2）**树深度**的数学期望值。约定根结点的深度为 0。

【输入格式】

输入仅有一行，包含两个正整数 q, n，分别表示问题编号以及叶结点的个数。

【输出格式】

输出仅有一行，包含一个实数 d，四舍五入精确到小数点后 6 位。如果 $q = 1$，则 d 表示叶结点平均深度的数学期望值；如果 $q = 2$，则 d 表示树深度的数学期望值。

【输入样例 1】

```
1 4
```

【输出样例 1】

```
2.166667
```

【输入样例 2】

```
2 4
```

【输出样例 2】

```
2.666667
```

【输入样例 3】

```
1 12
```

【输出样例 3】

```
4.206421
```

【输入样例 4】

```
2 12
```

【输出样例 4】

```
5.916614
```

【样例 1、样例 2 说明】

数学期望值是随机变量的值乘以其概率的总和：记随机变量 X 可能的取值为 $x_1, x_2, \ldots, x_n$，它们取到的概率分别为 $p_1, p_2, \ldots, p_n$，那么随机变量 X 的数学期望值就是

$$E(X) = \sum_{i=1}^{n} p_i x_i$$

例如，掷一枚写有 1、2、3、4、5、6 这 6 个数的均匀骰子，掷到的数的数学期望值是：

$$E = \frac{1}{6} \times 1 + \frac{1}{6} \times 2 + \frac{1}{6} \times 3 + \frac{1}{6} \times 4 + \frac{1}{6} \times 5 + \frac{1}{6} \times 6 = 3.5$$

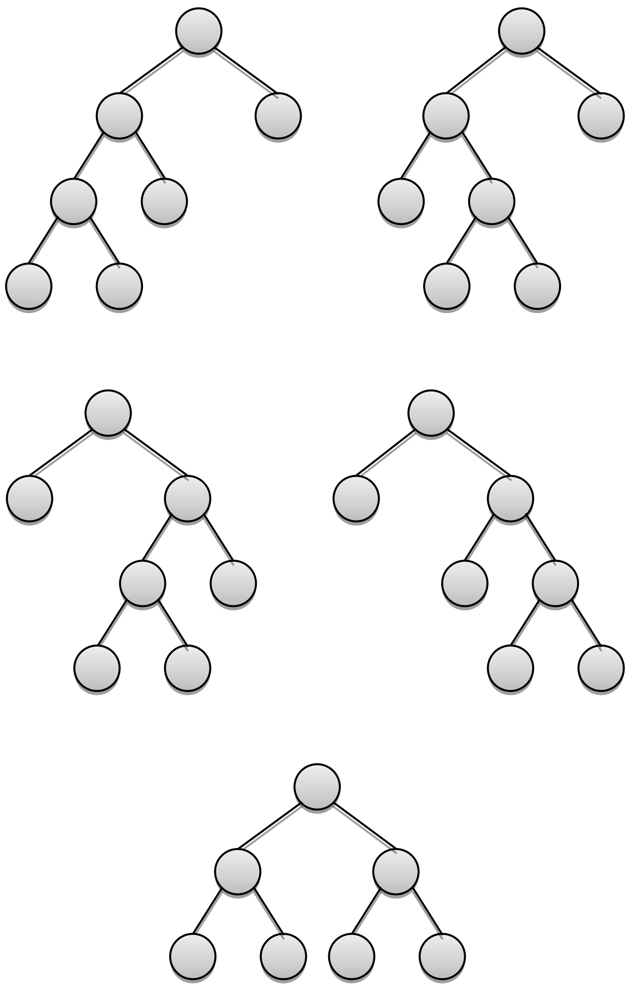
尽管 3.5 不是骰子上的某个数。又如，一道 4 选 1 的选择题，答对得 5 分，不答不得分，答错倒扣 1 分。那么，当我们不答时，一定得 0 分；而等概率地随便猜一个时，得分的数学期望值是：

本题中，根据二叉树的生成方式，当 $n = 4$ 时，下图中前四棵树被生成的概率均为 1/6，最后一棵树被生成的概率为 1/3。它们的**叶结点平均深度**分别是 9/4、9/4、9/4、9/4、2，因此**叶结点平均深度**的数学期望值是

$$E=\frac{1}{6}\times\frac{9}{4}+\frac{1}{6}\times\frac{9}{4}+\frac{1}{6}\times\frac{9}{4}+\frac{1}{6}\times\frac{9}{4}+\frac{1}{3}\times 2=\frac{13}{6}$$

而它们的**树深度**分别是 3、3、3、3、2，因此**树深度**的数学期望值是

$$E=\frac{1}{6}\times 3+\frac{1}{6}\times 3+\frac{1}{6}\times 3+\frac{1}{6}\times 3+\frac{1}{3}\times 2=\frac{8}{3}$$



【数据规模】

测试数据编号	q	n
1, 2	$q = 1$	$2 \le n \le 10$
3, 4, 5		$2 \le n \le 100$
6, 7	$q = 2$	$2 \le n \le 10$
8, 9, 10		$2 \le n \le 100$