

Problem A 钟的故事

The Story of Clock

输入文件: **clock.in**

钟是人们最普遍的计时工具。早在中世纪,钟就随着人们对机械力学的研究出现了。世界上第一只钟是用齿轮驱动的。它出现在公元 1386 年。

韩国人特别喜爱钟,有世界佛教国家中最大的钟,不但如此,韩国人还乐于在自己的名字中加入一个“钟”字。

钟的故事很精彩,不过这与本题无关,这里就不再赘述了。

一个普通的钟是通过时针、分针和秒针指向的圆盘刻度来表示时间的。三根针的一端都被固定在圆盘的中央。圆盘一周被均分成 12 大格,标有刻度 1 至 12,每一大格又被均分成 5 小格。时针读大格,分针和秒针读小格。分针每小时走一圈;时针每小时走一大格,半天(12 小时)走一圈;秒针每秒走一小格。每天子夜 0 点和正午 12 点,三根针都指向 12。理想的钟的三根针上的任何一点可以被认为是在做匀速圆周运动,且可以被认为是无限细的。

你现在的任务是:给你一个时刻,请给出该时刻钟的时针与分针的夹角。

输入:

输入文件给出了若干时刻,格式为: hh:mm:ss, 其中 $0 \leq hh \leq 23$, $0 \leq mm \leq 59$, $0 \leq ss \leq 59$, 分别表示小时,分钟和秒。

输出:

对于输入文件中的每一个时刻,用角度制给出该时刻钟的时针与分针的夹角。你输出的夹角必须在 0° 到 180° 之间(包括 0° 和 180°)。输出格式为: $D^\circ MM' SS''$, 即 D 度 MM 分 SS 秒, 其中 $0 \leq D \leq 180$, $0 \leq MM \leq 59$, $0 \leq SS \leq 59$ 。D 按实际长度输出。MM 和 SS 输出时占两位,不足两位时在数字前用 0 补足。

输入样例: (**clock.in**)

9:00:00
18:00:00
12:00:00

输出样例:

$90^\circ 00' 00''$
 $180^\circ 00' 00''$
 $0^\circ 00' 00''$

以下信息可能对你有帮助:

角度符号°的 ASCII 码是 248,
单引号'的 ASCII 码是 39,
双引号"的 ASCII 码是 34。

Problem B 红色病毒

Alert! Red Virus!!!

输入文件: **redvirus.in**

最近发现了一种新病毒，因为其蔓延速度与最近在 Internet 上传播的“红色代码”不相上下，故被称作“红色病毒”。经研究发现，该病毒及其变种的 DNA 的一条单链中，胞嘧啶、腺嘌呤均是成对出现的。这虽然是一个重大发现，但还不是该病毒的最主要特征，因为这个特征实在太弱。为了搞清楚该病毒的特征，韩国三星电子公司委派软件工程师 Grant 对此产生了兴趣。他想知道在这个特征下，可能成为病毒的 DNA 序列的个数。更精确的说，Grant 需要统计所有满足下列条件的长度为 n 的字符串的个数：

1. 字符串仅由 A,T,C,G 组成。
2. A 出现偶数次（也可以不出现）。
3. C 出现偶数次（也可以不出现）。

当 $n=2$ 时，所有满足条件的字符串有如下 6 个：

TT, TG, GT, GG, AA, CC。

由于这个数可能非常庞大，你只需给出最后两位数字即可。

输入：

输入文件给出了若干个 n ($1 \leq n \leq 10^9$)。最后以 0 结束。

输出：

对于输入文件中的每一个 n ，输出满足条件的字符串的个数的最后两位数字。

输入样例：(**redvirus.in**)

1
2
3
100
0

输出样例：

2
6
20
32

Problem C 多米诺效应

The Domino Effect

输入文件: **domino.in**

为了迎接新世纪，中日韩三国学生在北京进行了多米诺骨牌表演，创造了吉尼斯纪录。在多米诺骨牌表演中，时间的计算是一个关键环节。现在给定骨牌的摆放，请计算哪块骨牌最后倒下和所需时间。

整套多米诺系统由若干关键性的骨牌和连接它们的若干条骨牌组成。如果某个关键性的骨牌倒下，所有与之相邻的骨牌也将倒下（除了已经倒下的骨牌）。倒下的骨牌可能碰到其他关键性的骨牌从而引发连锁反应。一条骨牌可能从任一方向倒下，也可能从两边一起倒下。所以，最后倒下的骨牌不一定是关键性的骨牌，而处于两个关键性的骨牌之间。你可以假设骨牌是以恒定的速率倒下的。

输入：

输入文件给出了多组数据。每组数据的第一行由两个整数 n, m ($1 \leq n \leq 100$)。 n 表示关键性的骨牌个数， m 表示骨牌的条数，关键性的骨牌以 1 至 n 编号。接下来的 m 行，每行三个整数 a, b, t ， ($1 \leq a, b \leq n$, $0 \leq t \leq 10^7$)。表示骨牌 a 与骨牌 b 之间的骨牌全部倒下需要 t 秒。两个关键性的骨牌之间最多只有一条骨牌，也可能没有。所有骨牌都是相连的。我们如果推倒一个骨牌，所有骨牌最终都将倒下。在时刻 0，我们将推倒关键性的骨牌 1。

$n=m=0$ 时，输入文件结束。

输出：

对于每组数据，首先输出编号 (System #1, System #2 等等)。如果最后倒下的骨牌是关键性的骨牌，则输出：

The last domino falls after T seconds, at key domino A.

其中 T 是最后倒下骨牌倒下的时间， A 是那骨牌的编号。如果最后倒下的骨牌不是关键性的骨牌，则输出：

The last domino falls after T seconds, between key dominoes A and B.

其中 T 是最后倒下骨牌倒下的时间， A, B 是骨牌两端关键性的骨牌的编号， $A < B$ 。以上 T 均需保留一位小数。每组数据处理完之后请输出一个空行。

输入样例: (**domino.in**)

```
2 1
1 2 27
3 3
1 2 5
1 3 5
2 3 5
0 0
```

输出样例：

System #1

The last domino falls after 27.0 seconds, at key domino 2.

System #2

The last domino falls after 7.5 seconds, between key dominoes 2 and 3.

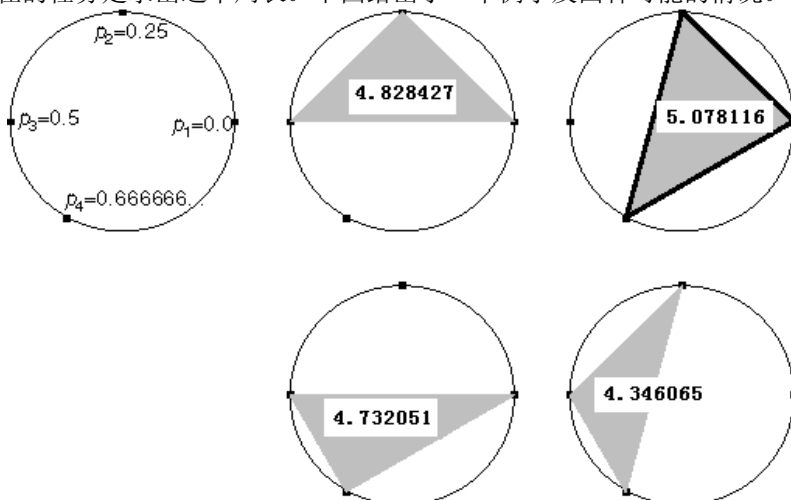
Problem D 法老之谜

The Mystery of Pharaoh

输入文件: **pharaoh.in**

埃及的金字塔是一个神秘又充满危险的地方。著名的“三星”探险队为了揭开许多不为人知的秘密，又深入了法老王的墓地。探险队闯过了一道又一道机关，接近了墓地的核心。此时，又一道机关出现了。如果不能闯过这道机关，就意味着可能前功尽弃。“三星”探险队决不会允许这样的事情发生。

幸运的是，队中的古埃及文化专家弄明白了机关的含义。摆在面前的是一个半径为 1 的圆。在圆周上分布着 n 个开关。需要开启其中 m 个开关。探险队从法老的经文得到暗示，由这 m 个开关组成的凸多边形的周长一定是所有 m 个开关组成的凸多边形周长中最长的。你现在的任务是求出这个周长。下图给出了一个例子及四种可能的情况。



输入:

输入文件给出了若干组数据，每组数据的第一行为两个整数 n, m ($3 \leq m \leq n \leq 40$)。第二行有 n 个实数 $0 \leq p_1 < p_2 < \dots < p_n < 1$ ，其中 p_i 表示由圆心指向第 i 个开关的向量的幅角主值为 $2 * p_i * \pi$ 。

$n=m=0$ 时，输入文件结束。

输出:

对于每组输入数据，输出你求得的最大周长。周长须保留 6 位小数。

输入样例: (**pharaoh.in**)

```
4 4
0 0.25 0.5 0.75
4 3
0 0.25 0.5 0.6666666666666667
0 0
```

输出样例:

```
5.656854
5.078116
```

Problem E 销售计划

The Selling Plan

输入文件: **sell.in**

三星电子作为三星集团的三个子公司之一,以电子高科技领域的新产品研制开发而闻名世界。三星电子相继开发出世界上第一个 1G、4G 动态存储器,成功开发了世界上第一个 22 英寸和 30 英寸 TFT-LCD 液晶显示器。三星电子的其它数字电子产品同样很有名,这些产品的销售遍及世界各地。

为了更好地做好销售工作,三星上海市的销售主管制定了一个所谓的“销售线”的计划:所有的销售点分布在城市的平面图上,在图上自左向右连接两销售点成线段,注意这些线段与水平 X 轴的夹角总在正负 45 度(包括正负 45 度)之间,连续不间断的相连线段成为一条销售线。你的任务是:在给定的销售点上如何连接,使得销售线的总数最少?

输入:

输入文件含有多组数据。每组数据的第一行为正整数 n ($n \leq 30000$),表示共有多少个销售点。以后的 n 行中的每一行为正整数 x 和 y ($0 \leq x, y \leq 30000$),表示销售点的坐标。输入文件的最后一行为 -1,表示结束。

输出:

共有 n 行,每行一个整数,表示对应一组数据的最少销售线。

输入样例: (**sell.in**)

```
2
1 1
2 2
6
1 6
10 8
1 5
2 20
4 4
6 2
-1
```

输出样例:

```
1
3
```