

# 信息工程学院

## 程序设计大赛试题

(2012.5)



---

### 友情提示:

1. 本次竞赛共 7 题，竞赛时间从 18:30 到 21:30，共计 3 小时。
2. 题目的难度与题目的顺序、长度之间并无直接关系，最后一题为英文题，选手可以使用纸质词典。
3. 比赛期间请不要使用手机等电子通讯设备；队员可以在小组内部讨论，队与队之间禁止传播信息，否则按作弊处理。
4. 题目对输出格式有着严格的要求，每个题目的测试数据不仅仅局限于样例，请仔细调试检查后再进行提交，以避免不必要的时间损失。
5. 题目输出时用到的所有符号均为英文半角格式。
6. 比赛期间有任何问题请举手示意，除裁判外，现场工作人员不会对题目的问题作任何回答。
7. 所有题目的最终解释权归编程爱好者协会所有。

### 目录

1001.计算面积.....	2
1002.苹果零售店 .....	3
1003.尼科彻斯定理.....	4
1004.行编辑程序 .....	5
1005.素数环问题 .....	6
1006.霍格沃兹的火车 .....	7
1007.Tom and Jerry.....	8

## 1001.计算面积

### ➤ Description

计算如下的抛物线被直线所截阴影部分的面积（见图 1001），其中 P1 是抛物线的顶点。

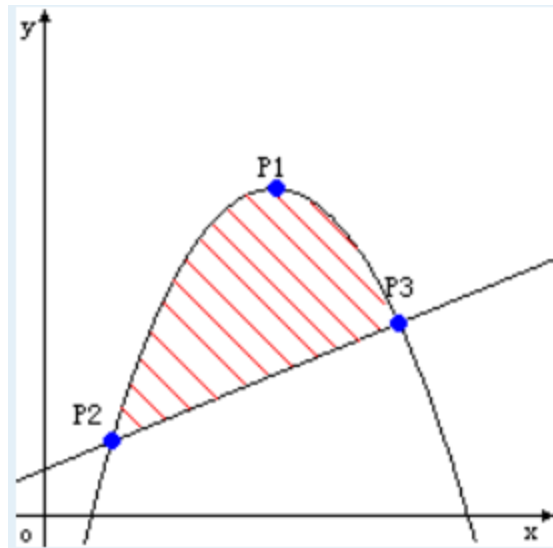


图 1001 抛物线

### ➤ Input

首先输入一个整数  $n$  ( $0 < n < 10$ ), 表示测试数据的组数; 然后输入这  $n$  组测试数据, 每组测试数据都包括 3 行, 分别为 P1,P2,P3 这三个点的坐标, 所有坐标值都是不大于 1000.0 的非负数。

### ➤ Output

输出包括  $n$  行, 每行为对应于该行测试数据下阴影部分的面积, 保留两位小数。

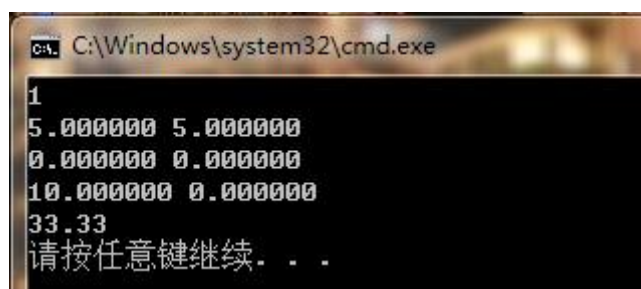
### ➤ Sample Input

```
1
5.000000 5.000000
0.000000 0.000000
10.000000 0.000000
```

### ➤ Sample Output

```
33.33
```

### ➤ 附测试截图:



## 1002.苹果零售店

### ➤ Description

我们都知道苹果公司是世界上市值最多的公司，旗下拥有众多风靡世界的电子产品，如 iPhone、iPad、Mac Book Air 等（见图 1002）。

现在苹果零售店的员工面临一个问题：在不使用任何辅助设备的情况下，如何找出一天之内该零售店售出的所有产品中数量最多的那一种，请你来帮助他们。

### ➤ Input

输入包含多组测试数据，对于每组测试数据的个数，首先输入一个  $N$  ( $0 < N \leq 1000$ ) 表示该组测试数据的个数，接下来输入这  $N$  个测试数据，为了简单起见，我们把苹果公司的所有产品都合成一个单词（例如，“MacBook Air”合成为“MacBookAir”）。请注意，表示产品的单词最长不超过 15 个英文字母。如果输入的  $N$  为 0 则统计结束。

### Output

输出  $N$  行信息，每行为对应组的测试数据统计的结果，输出该组中数量最多的产品。每组测试数据中，数量最多的产品不会多于一种。

### ➤ Sample Input

```
5
iPhone
iPad
iPod
iPhone
iPhone
3
MacBookAir
MacBookPro
MacBookAir
0
```

### ➤ Sample Output

```
iPhone
MacBookAir
```

### ➤ 附测试截图：



图 1002 苹果 iPad

```
C:\Windows\system32\cmd.exe
5
iPhone
iPad
iPod
iPhone
iPhone
iPhone
3
MacBookAir
MacBookPro
MacBookAir
MacBookAir
0
请按任意键继续. . .
```

## 1003.尼科彻斯定理

### ➤ Description

尼科彻斯定理是数学上的一个有趣的定理，这个定理的基本内容是：任何一个整数的立方都可以写成一串连续奇数的和。比如，对于整数 13，按照题目中关于尼科彻斯定理的定义，可以用等式将它表示成如下的形式：  
 $13 \times 13 \times 13 = 2197 = 157 + 159 + 161 + 163 + 165 + 167 + 169 + 171 + 173 + 175 + 177 + 179 + 181$ 。

本题是一个定理，我们先来证明它是成立的。

对于任一正整数  $a$ ，不论  $a$  是奇数还是偶数，整数  $(a \times a - a + 1)$  必然为奇数。

构造一个等差数列，数列的首项为  $(a \times a - a + 1)$ ，等差数列的差值为 2 (奇数数列)，则前  $a$  项的和为：

$$\begin{aligned} & a \times ((a \times a - a + 1) + 2 \times (a - 1) / 2) \\ &= a \times a \times a - a \times a + a + a \times a - a \\ &= a \times a \times a \end{aligned}$$

定理成立。证毕。

现在请你通过编程验证尼科彻斯定理。

### ➤ Input

首先输入一个整数  $n$  ( $n < 10$ )，表示测试数据的组数；然后输入这  $n$  个测试数据，每个测试数据都不大于 100。

### ➤ Output

输出包括  $n$  行，每行的信息为该行对应的测试数据按照题目叙述表示成尼科彻斯定理的形式（具体形式参见样例）。

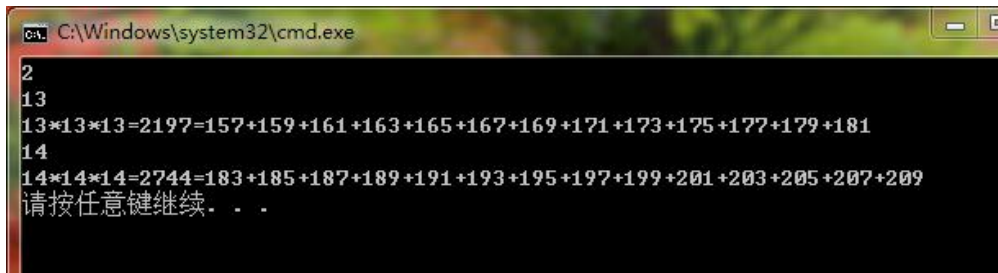
### ➤ Sample Input

```
2
13
14
```

### ➤ Sample Output

```
13*13*13=2197=157+159+161+163+165+167+169+171+173+175+177+179+181
14*14*14=2744=183+185+187+189+191+193+195+197+199+201+203+205+207+209
```

➤ 附测试截图：



```
C:\Windows\system32\cmd.exe
2
13
13*13*13=2197=157+159+161+163+165+167+169+171+173+175+177+179+181
14
14*14*14=2744=183+185+187+189+191+193+195+197+199+201+203+205+207+209
请按任意键继续. . .
```

## 1004.行编辑程序

➤ **Description**

一个简单的行编辑程序的功能是：接受用户从终端输入的程序或数据，并存入用户的数据区。由于用户在终端上进行输入时，不能保证不出差错，因此，若在编辑程序中，“每接受一个字符即存入用户数据区”的做法显然不是最恰当的。较好的做法是，设立一个输入缓冲区，用以接受用户输入的一行字符，然后逐行存入用户数据区。允许用户输入出差错，并在发现有误时可以及时更正。例如，当用户发现刚刚键入的一个字符是错的时，可补进一个退格符“#”，以表示前一个字符无效；如果发现当前键入的行内差错较多或难以补救，则可以键入一个退行符“@”，以表示该字符之前的字符均无效。

例如，假设从终端接受了这样的两行字符：

```
whi#ilr#e(s#*s)
```

```
outcha@putchar(*s=#++);
```

则实际有效的是下列两行：

```
while(*s)
```

```
putchar(*s++ );
```

为了简便起见，我们假定从终端输入的每行字符中最多只有一个“@”，并且不会出现相邻的两个“#”（比如像 abcd##cde###f 这类的字符串就不会出现）。现在请你编程解决上述问题。

➤ **Input**

首先输入一个整数  $n$  ( $n < 10$ )，表示测试数据的组数；然后输入这  $n$  个测试数据，每个测试数据的长度都不大于 100。

➤ **Output**

输出包括  $n$  行，每行分别为对应的测试数据采用行编辑程序还原之后的字符串，如果还原后没有一个字符，则该行将成为空行。为美观起见，我们规定凡空行则输出下列引号中的字符串：“####@###”。

➤ **Sample Input**

3

```
whi#ilr#e(s#*s)
```

```
outcha@putchar(*s=#++)
```

```
a#b#c@g#
```

➤ **Sample Output**

```
while(*s)
putchar(*s++ )
###@###
```

➤ 附测试截图：

```
C:\Windows\system32\cmd.exe
3
whi#ilr#e(s#*s)
while(*s)
outcha@putchar(*s=#++)
putchar(*s++)
a#b#c@g#
###@###
请按任意键继续. . .
```

## 1005.素数环问题

### ➤ Description

常见的对于素数的定义是：素数，又称为质数，是不能被 1 与它本身以外的其它整数整除的整数。如 2, 3, 5, 7, 11, 13, 17 是前几个质数，其中 2 为唯一的偶素数。

现在，我们的问题是，给定一个由  $n$  个圆圈组成的环状结构，请你将  $1 \sim n$  这  $n$  个整数分别填入这  $n$  个圆圈中，使得任意相邻的两个圆圈中的数字之和均为素数（样例请参照图 1005）。请注意，在填充时，数字 1 总是第一个填入的。

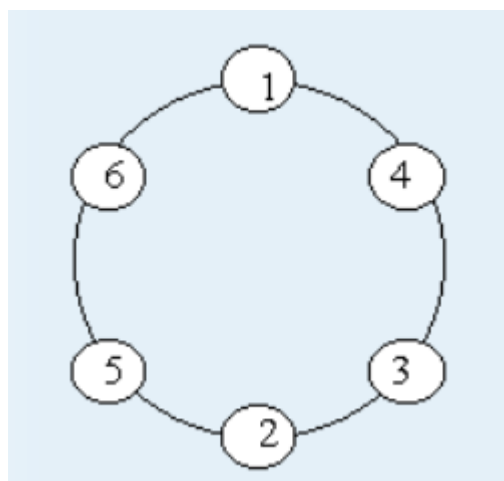


图 1005 素数环

### ➤ Input

输入包括多个正整数  $n$  ( $1 < n < 20$ )，输入的  $n$  值为 0 表示输入结束。

### ➤ Output

输出的格式请参照样例。对于输入每个  $n$  值，先输出 “Case k:”， $k$  表示样例的编号，然后按照字典序输出这  $n$  个数的所有可能组合，在每一个  $n$  值对应的所有输出结果全部结束后有一个换行。请注意：对于同一种填充方案，顺时针和逆时针是 2 种不同的输出组合。输入数据保证都至少有一种填充方案。

### ➤ Sample Input

```
6
8
0
```

### ➤ Sample Output

```
Case 1:
1 4 3 2 5 6
1 6 5 2 3 4
```

Case 2:

1 2 3 8 5 6 7 4

1 2 5 8 3 4 7 6

1 4 7 6 5 8 3 2

1 6 7 4 3 8 5 2

➤ 附测试截图:

```
C:\Windows\system32\cmd.exe
6
Case 1:
1 4 3 2 5 6
1 6 5 2 3 4

8
Case 2:
1 2 3 8 5 6 7 4
1 2 5 8 3 4 7 6
1 4 7 6 5 8 3 2
1 6 7 4 3 8 5 2

0
请按任意键继续. . .
```

## 1006.霍格沃兹的火车

### ➤ Description

霍格沃兹魔法学校又如期开学啦！这一次哈利的两个儿子阿不思·西弗勒斯·波特、詹姆·小天狼星·波特和他的同伴们遇到了一个棘手的问题：原来宽阔的九又四分之三车站正在进行大规模的改建，只留有一个站口供进站和出站。而由此导致的一个严重问题就是车站一次只能出入一辆火车，也就是说，当 A 车进站后，如果 B 车紧跟着进站，那么只有当 B 车出站后，A 车才能跟着出站（请参照图 1006）。哈利那天生聪明的女儿莉莉·卢娜·波特出了一个难题，她的问题是，任意给定一系列车次的进站序列和出站序列，然后让阿不思·西弗勒斯·波特和詹姆·小天狼星·波特以及他们的朋友来判断，在相应的进站序列下，所有的车次能否按照给出的出站序列全部出站。答对者会获得一份莉莉·卢娜·波特送出的一份神秘礼物，据说是哈利·波特单独传授她的绝密魔法哦！请你帮助他们获得这份礼物。

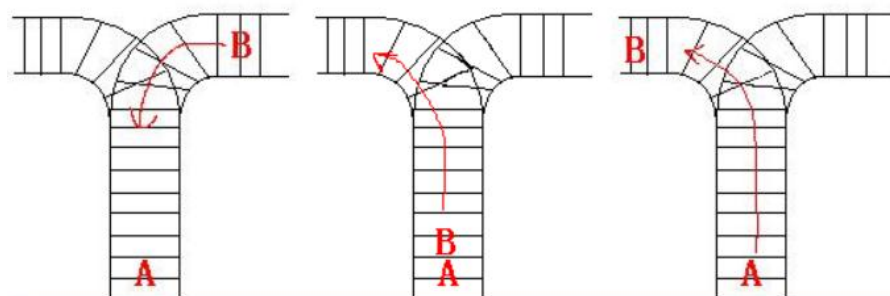


图 1006 火车出入站示意



➤ **Input**

输入包括多行测试数据。首先输入一个正整数  $m$ ，表示测试数据的组数，接着输入一个正整数  $n$  ( $n \leq 9$ )，表示车次的总数，然后输入两个序列  $O1$  和  $O2$ ，分别表示入站及出站序列。当  $n$  的值为 0 时表示输入结束。

➤ **Output**

当以  $O1$  入站的车能以  $O2$  的顺序全部出站时，先输出一个 **Yes**，然后输出这  $n$  行车站的调度信息，这  $n$  行调度信息是由 **in** 和 **out** 组成的：对于车站来说，只有两种状态，**in** 表示当前的车次进站，**out** 则表示出站。如果不能全部出站，则只输出一个 **no**。在一个测试数据输出结束后，输出一个 **FINISH** 表示样例结束，同时换行。

➤ **Sample Input**

```
2
3 123 321
3 123 312
```

➤ **Sample Output**

```
Yes.
in
in
in
out
out
out
FINISH
No.
FINISH
```

➤ 附测试截图：



```
C:\Windows\system32\cmd.exe
2
3 123 321
Yes.
in
in
in
out
out
out
FINISH
3 123 312
No.
FINISH
请按任意键继续. . .
```

## 1007.Tom and Jerry

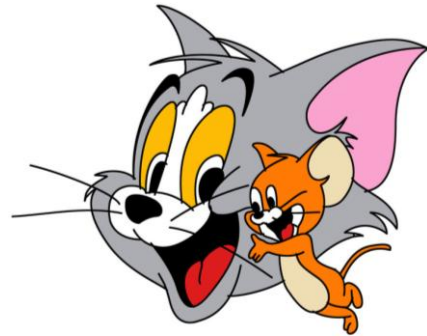
➤ **Description**

In the famous carton *Tom and Jerry*, Tom is a blue and white domestic shorthair cat. He is the protagonist of the story, who usually lives a pampered life, although



they usually live in several lifestyles, while Jerry is a small brown house mouse who always lives in close proximity to him and is the winner of the story(please refer to Pic. 1007).

"Tom" is a generic name for a male cat. Tom was seen originally in the very first short, Puss Gets the Boot, and Jerry was seen in the short also, although it was not billed as a Tom and Jerry cartoon. Jerry possesses surprising strength for his size, lifting items such as anvils with relative ease and withstanding considerable impacts with them. Despite the typical cat-eats-mouse scenario, it is surprisingly quite rare for Tom to actually try and consume Jerry. Most of his attempts are just to torment or humiliate Jerry. Despite being very energetic and determined, Tom is no match for Jerry's brains and wits. By the final "fade-out" of each cartoon, Jerry usually emerges triumphant, while Tom is shown as the loser.



Pic.1007 Tom and Jerry

Today comes a moment in which we probably feel like assisting Jerry to play with Tom. The situation is quite simple: Both Tom and Jerry love eating their favorite foods, for example, cat food is what Tom prefers while Jerry is interested in JavaBean. Whereas all the JavaBeans are placed in separated rooms guarded by Tom. In order to get Tom out of the door, Jerry prepared  $M$  pounds of cat food which can be used to trade with Tom. There are  $N$  rooms in the house. The  $i$ -th room contains  $J[i]$  pounds of JavaBeans and requires  $F[i]$  pounds of cat food given to Tom. Jerry does not have to trade for all the JavaBeans in the room, instead, he may get  $J[i] * a\%$  pounds of JavaBeans if he pays  $F[i] * a\%$  pounds of cat food. Here,  $a$  is a real number. Now he is assigning this homework to you: tell him the maximum amount of JavaBeans he can obtain.

➤ **Input**

The input consists of multiple test cases. Each test case begins with a line containing two non-negative integers  $M$  and  $N$ . Then  $N$  lines follow, each contains two non-negative integers  $J[i]$  and  $F[i]$  respectively. The last test case is followed by two -1's. All integers are not greater than 1000.

➤ **Output**

For each test case, print in a single line a real number accurate up to 3 decimal places, which is the maximum amount of JavaBeans that Jerry can obtain.

➤ **Sample Input**

```
5 3
7 2
4 3
5 2
20 3
25 18
24 15
15 10
```

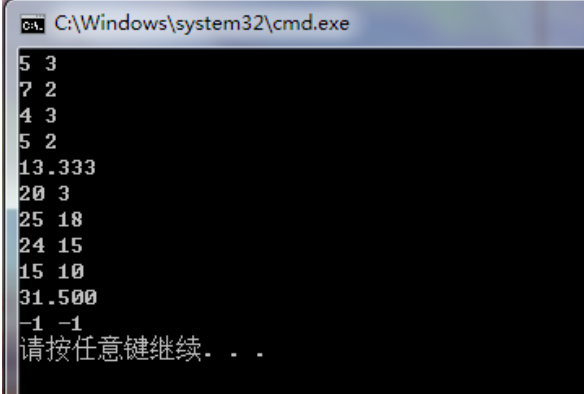
-1 -1

➤ **Sample Output**

13.333

31.500

➤ **附测试截图：**



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a black background with white text. The output of the program is as follows:

```
5 3
7 2
4 3
5 2
13.333
20 3
25 18
24 15
15 10
31.500
-1 -1
请按任意键继续. . .
```

©2012 年 5 月，编程爱好者协会。