## 基本测试试题

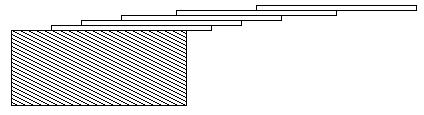
# 算法部分：

## 基本要求：可以任选二个题（多不限），可以用伪代码实现，但必须描述清楚解题的思路

## 算法一：

## 描述

How far can you make a stack of cards overhang a table? If you have one card, you can create a maximum overhang of half a card length. (We're assuming that the cards must be perpendicular to the table.) With two cards you can make the top card overhang the bottom one by half a card length, and the bottom one overhang the table by a third of a card length, for a total maximum overhang of 1/2 + 1/3 = 5/6 card lengths. In general you can make *n*cards overhang by 1/2 + 1/3 + 1/4 + ... + 1/(*n* + 1) card lengths, where the top card overhangs the second by 1/2, the second overhangs tha third by 1/3, the third overhangs the fourth by 1/4, etc., and the bottom card overhangs the table by 1/(*n* + 1). This is illustrated in the figure below.



**输入**

The input consists of one or more test cases, followed by a line containing the number 0.00 that signals the end of the input. Each test case is a single line containing a positive floating-point number c whose value is at least 0.01 and at most 5.20; c will contain exactly three digits.

**输出**

For each test case, output the minimum number of cards necessary to achieve an overhang of at least c card lengths. Use the exact output format shown in the examples.

**样例输入**

1.00

3.71

0.04

5.19

0.00

**样例输出**

3 card(s)

61 card(s)

1 card(s)

273 card(s)

## 算法二：

## 描述

一个工厂制造的产品形状都是长方体，它们的高度都是h，长和宽都相等，一共有六个型号，他们的长宽分别为1\*1, 2\*2, 3\*3, 4\*4, 5\*5, 6\*6。这些产品通常使用一个 6\*6\*h 的长方体包裹包装然后邮寄给客户。因为邮费很贵，所以工厂要想方设法的减小每个订单运送时的包裹数量。他们很需要有一个好的程序帮他们解决这个问题从而节省费用。现在这个程序由你来设计。

**输入**

输入文件包括几行，每一行代表一个订单。每个订单里的一行包括六个整数，中间用空格隔开，分别为1\*1至6\*6这六种产品的数量。输入文件将以6个0组成的一行结尾。

**输出**

除了输入的最后一行6个0以外，输入文件里每一行对应着输出文件的一行，每一行输出一个整数代表对应的订单所需的最小包裹数。

**样例输入**

0 0 4 0 0 1

7 5 1 0 0 0

0 0 0 0 0 0

**样例输出**

2

1

## 算法三：动态规划

## 描述

Michael喜欢滑雪百这并不奇怪， 因为滑雪的确很刺激。可是为了获得速度，滑的区域必须向下倾斜，而且当你滑到坡底，你不得不再次走上坡或者等待升降机来载你。Michael想知道载一个区域中最长的滑坡。区域由一个二维数组给出。数组的每个数字代表点的高度。下面是一个例子

1 2 3 4 5  
16 17 18 19 6  
15 24 25 20 7  
14 23 22 21 8  
13 12 11 10 9

一个人可以从某个点滑向上下左右相邻四个点之一，当且仅当高度减小。在上面的例子中，一条可滑行的滑坡为24-17-16-1。当然25-24-23-...-3-2-1更长。事实上，这是最长的一条。

**输入**

输入的第一行表示区域的行数R和列数C(1 <= R,C <= 100)。下面是R行，每行有C个整数，代表高度h，0<=h<=10000。

**输出**

输出最长区域的长度。

**样例输入**

5 5

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

**样例输出**

25

## 算法四：搜索

**描述**

在一个给定形状的棋盘（形状可能是不规则的）上面摆放棋子，棋子没有区别。要求摆放时任意的两个棋子不能放在棋盘中的同一行或者同一列，请编程求解对于给定形状和大小的棋盘，摆放k个棋子的所有可行的摆放方案C。

**输入**

输入含有多组测试数据。  
每组数据的第一行是两个正整数，n k，用一个空格隔开，表示了将在一个n\*n的矩阵内描述棋盘，以及摆放棋子的数目。 n <= 8 , k <= n  
当为-1 -1时表示输入结束。  
随后的n行描述了棋盘的形状：每行有n个字符，其中 # 表示棋盘区域， . 表示空白区域（数据保证不出现多余的空白行或者空白列）。

**输出**

对于每一组数据，给出一行输出，输出摆放的方案数目C （数据保证C<2^31）。

**样例输入**

2 1

#.

.#

4 4

...#

..#.

.#..

#...

-1 -1

**样例输出**

2

1

# Java部分：

(1) 自己编程实现从网上抓一个网页  
(2) 将网页所有的html相关标签去掉，生成一个纯文本文件，保存为文件  
(3) 对(2)生成的文本文件以字为单位进行分析，列出出现次数最多的前10个字和相应的次数

# Hadoop部分：

(1)实现一个map reduce编程，完成一个多表关联的操作：

如输入文件a：

Beijing red 722

Shanghai green 743

Shenzhen blue 755

文件b：

733 liming

722 sandy

743 litao

输出：

Beijing red 722 sandy

Shanghai green 743 litao

(2)为什么在hadoop中自己实现了一套序列化框架而不是用java自己的序列化机制，试着从原理和源码级进行分析，并提供具体的测试代码级样例。