

E4-Statement

助教们的贴心寄语

清明节3天假期，天气这么好，为什么不出门玩一玩呢？

比赛详情

- 练习时间2022/04/02 08:00 - 2022/04/13 18:00
- 总共11道题，编号为A~K，**题目不一定按难度排序**，同学们可以按照通过人数答题；或者提前读完所有题目，按照自己擅长题目的情况答题。
- 本次比赛的最后三道题目是C5的预习题，预习题计1分。
- 各题分值分布为如下，总分100+3分：

序号	A	B	C	D	E	F	G	H	I	J	K
分值	20	20	20	15	10	5	5	5	1	1	1

- 请严肃练习，严禁抄袭他人代码，课程组会在练习结束后进行代码查重并给予警告。

组题：呱呱泡蛙

更多题目作者：cbj、JJJ、wwh、LNB、LJF、Arthas、zym、魔法少女zhn、czy。

A 二进制转十进制(水题速来)

题目描述

把一个二进制数转换为十进制非负整数。

输入

一个小于等于64位的二进制数。

输出

一个十进制非负整数（保证范围在 `unsigned long long` 之内）。

输入样例

输出样例

1 13

author:cbj

B WWH函数

题目背景

某一天，*wwh*看到威廉·阿克曼发明了阿克曼函数的壮举，他突发奇想，写出了一个WWH函数，但是他不会算，请同学们帮帮他。

题目描述

WWH 函数 $solve(a, b, c)$ 中 a, b, c 定义域为 int 范围内的整数，函数值定义为：

$$solve(a, b, c) = \begin{cases} solve(8, 8, 8) & (a > 8 \text{ 或 } b > 8 \text{ 或 } c > 8 \text{ 时}) \\ 1 & (a \leq 0 \text{ 或 } b \leq 0 \text{ 或 } c \leq 0 \text{ 时}) \\ solve(a, b, c-1) + solve(a, b-1, c-1) & \\ -solve(a, b-1, c) & (0 < a, b, c < 8, \text{ 并且 } a < b < c \text{ 时}) \\ solve(a-1, b, c) + solve(a-1, b-1, c) & \\ +solve(a-1, b, c-1) - solve(a-1, b-1, c-1) & (\text{其他情况时}) \end{cases}$$

输入格式

三个 int 范围内的整数 a, b, c ，中间用空格隔开。

输出格式

一个整数，代表WWH函数 $solve(a, b, c)$ 的值，数据保证存在输出结果。

输入样例

```
1 2 2 2
```

输出样例

```
1 4
```

AUTHOR : wwh

C 简单的数学题

题目介绍

正如题目所说，这道题是一道数学题。

*JJJ*一共有 n 盏灯，一开始所有的灯都是**关**的。

*JJJ*会进行 n 次操作，第1次操作会把编号为1的倍数的灯状态进行改变（即开的变成关的，关的变成开的），第2次操作会把编号为2的倍数的灯状态进行改变，第3此操作会把3的倍数的灯.....

进行完 n 次操作后，有一些灯是**开**的，输出这些灯的编号。

输入格式

一个正整数 n ，表示灯的数目，以及操作的次数。

输出格式

一行若干个正整数，表示最后开着的灯的编号。

输入样例1

```
1 5
```

输出样例1

```
1 1 4
```

数据范围

对于40%的数据， $1 \leq n \leq 1000$;

对于100%的数据， $1 \leq n \leq 10^9$.

HINT

0: 关; 1: 开。

初始: 0 0 0 0 0

第一次: 1 1 1 1 1

第二次: 1 0 1 0 1

第三次: 1 0 0 0 1

第四次: 1 0 0 1 1

第五次: 1 0 0 1 0

最后只有1号和4号灯是开着的。

后记

不知道大家喜不喜欢这种思考多代码少的题目，反正我挺喜欢的。

Author: JJJ.

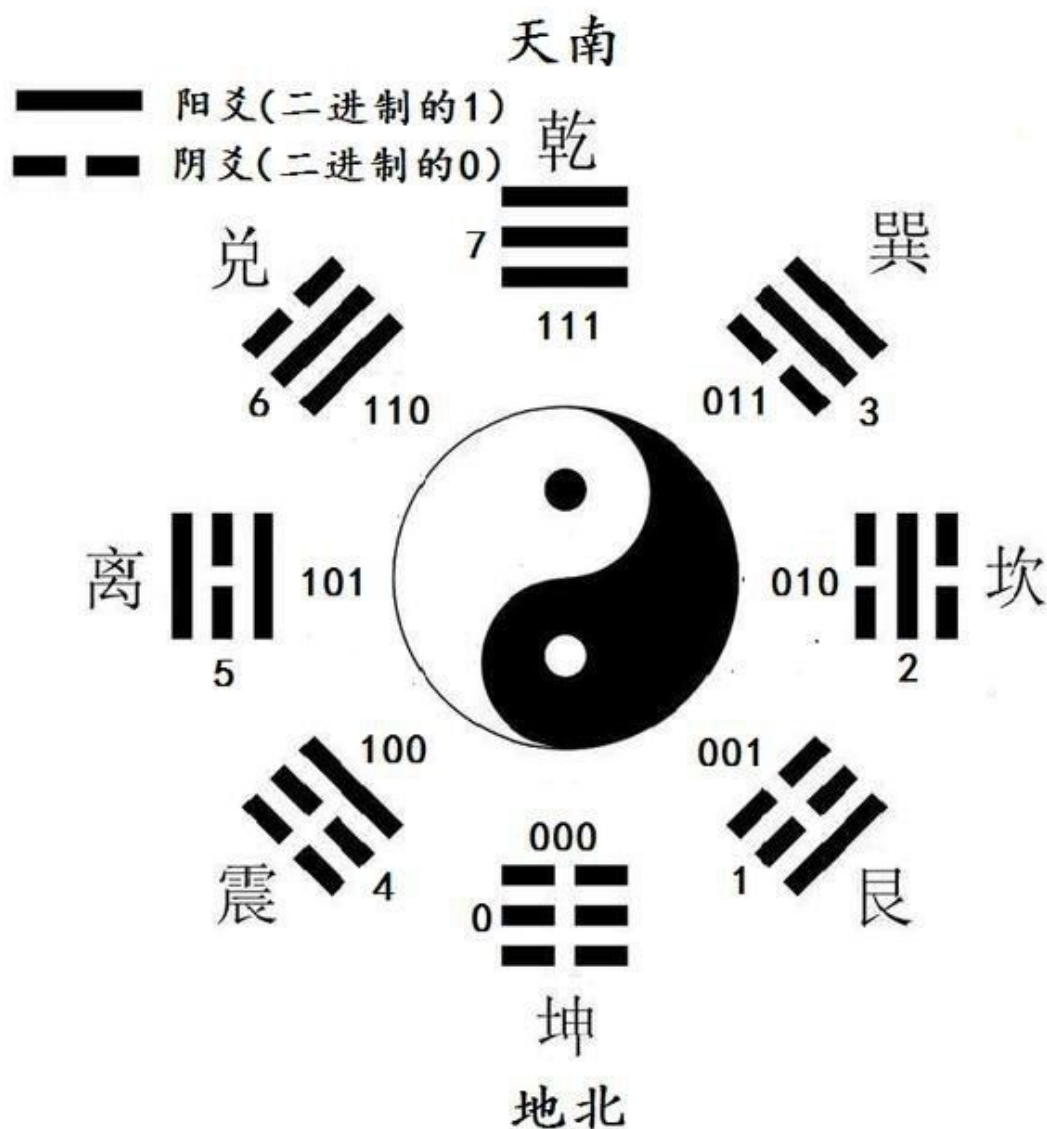
D 千年后的回响

题目描述

早在千年之前，古人基于对物质资料生产生活的观察，发现了这样的规律：天地、男女、昼夜、炎凉、上下、胜负……几乎生活中的一切现象都反映着**普遍的、相互对立的矛盾**，依照着这种朴素的经验与认识，我们的先祖把宇宙间变化多端、繁复万杂的事物分为**阴、阳两大类**，并用两种符号加以表示，阴物为— —，阳物为— — —，象征着广泛对立的种种事物与现象。

在此基础上，古人以阴、阳符号为爻，每三爻叠成一卦，出现了八卦，并发展到对自然界八种基本物质的具体象征，并在每卦的相互联系中披露了该卦所蕴涵的事物**运动、变化与发展**的哲理。八卦将任何事物都分成**对立统一**的关系加以阐释，将无穷的变化纳入这个数理的演绎体系中，并进而深化与拓展着人们对世界的本质特征与发展规律的认识。

这种用**组合的阴阳爻符号表示特定信息的方法**中贯穿着二进制编码的思想。我们如果将阴爻视作0，阳爻视作1，进而用八种不同的状态与二进制数码——建立联系，即用三位二进制数对八卦进行编码。在不知不觉中我们已经建立了一个简易的数制系统，换言之用一组固定的符号和统一的规则来表示数值。按照这种方式，万千的数字都可以使用八卦表达与演算。下图为八卦的简易编码系统。



若将八卦视作一个八进制数制系统，现约定以下概念：对一个数字 a ，需要 N 个八卦进行表示，约定其首个八卦为其本征卦，（即转换成八进制数后第0位所对应的卦），并约定卦的规模为 N （即其八进制数的总位数）。卦中阴阳爻的长度与其规模相关，设卦的规模为 N ，其阳爻使用 $3N$ 个短横线表示，其阴爻使用 $2N$ 个短横线与 N 个空格表示。

LNB学习了进制转换、循环等知识，希望对一个特定的数字 a 输出特定的卦形，即其规模为 N 的本征卦。

输入

一个十进制正整数 a 。（int范围内）

输出

输出三行，即对应的卦形。

输入样例

输出样例

```
1  -----
2  -----
3  ---    ---
```

样例解释

102的八进制为146，其第0位为6，本征卦对应为兑卦，规模为3。

Hint

对于重复性的操作比如输出阳爻或阴爻或许可以使用函数进行实现。

Author:LNB

E LJF算可靠度

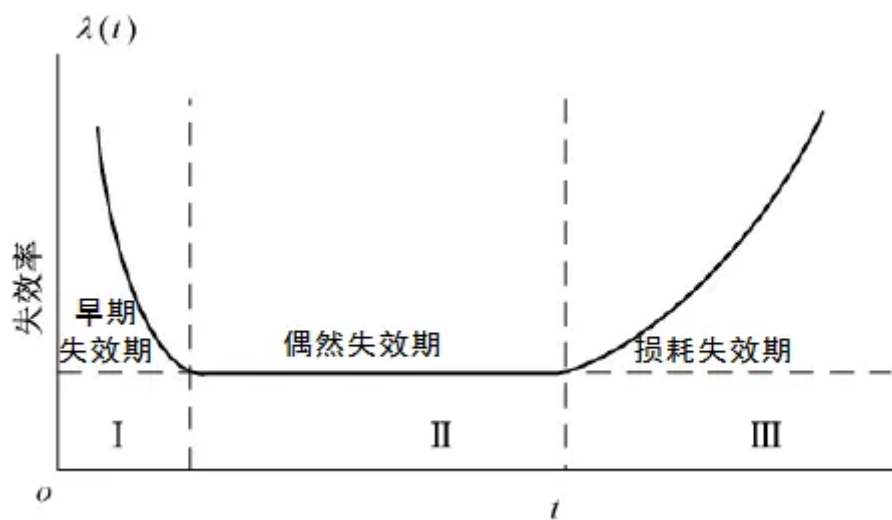
题目描述

经过一段时间的观察，*LJF*觉得校车晚到不一定是校车司机睡过了，很有可能是校车出故障了，来自**可靠性与系统工程学院**的*LJF*怎么能允许这种情况发生呢！所以*LJF*打算计算一下校车的可靠度。

可靠度计算原理如下：

产品的可靠度 $R(t) = e^{-\int_0^t \lambda(u) du}$ ， $\lambda(u)$ 为失效率关于时间变化的函数

下方为失效率函数的常见形式，因其很像浴盆，所以也被称为**浴盆曲线**。第一阶段是**早期失效期** (Infant Mortality)：表明产品在开始使用时，**失效率**很高,但随着产品工作时间的增加,失效率迅速降低,这一阶段失效的原因大多是由于设计、原材料和制造过程中的缺陷造成的。为了避免早期故障，一般会使用**老炼**的方式暴露早期故障，进入第二阶段**偶然失效期**后再交付使用。



现给出早期失效期的失效率函数 $\lambda(t) = e^{-(t+0.5)^2}$, $0 \leq t \leq 1.5$

聪明的你一看这不是一个普通的积分，能否利用**伪随机数**求出对应积分

输入

一行一个浮点数 t ，即可靠度公式中的积分上限。

输出

一个6位浮点数表示 t 时刻的可靠度，和标准值相差不超过0.001即算正确。

输入样例

```
1 1
```

输出样例

```
1 0.596348
```

数据范围

$$0 \leq t \leq 1.5$$

题目描述

北航高等理工学院19级有一套复杂的排名计算方案：

$$\text{总成绩} = \frac{GPA}{\text{年级最高}GPA} \times 50 + \frac{\text{荣誉课加权平均分}}{\text{年级最高荣誉课加权平均分}} \times 30 + \frac{\text{总学分数}}{\text{年级最高总学分数}} \times 20$$

那么，现在给你一份有若干个同学的成绩表，你能算出每位同学在这种评价体系下的总成绩吗？

注：

- “荣誉课”为高等理工学院一种特殊课种，大体等效为航类的置课。
- “加权平均分”中的权重是该课程的学分数。
- 北航的GPA算法为 $score \geq 60$ 时， $GPA = 4 - 3 \times \frac{(100 - score)^2}{1600}$ ，否则为0（ $score$ 为分数，总GPA计算方式为：先算出所有课程（包含所有荣誉课与非荣誉课）单独的GPA，再按学分求加权平均）。
- 这一批同学中的最高指标可作为年级的最高指标计算。

输入

第一行输入一个数 n ，代表这一批学生的人数。

接下来 n 组数据，代表 n 个学生，每一组中：

- 第一行输入两个数 id 和 m ，代表该学生的学号和选课的数量。
- 接下来 m 行，每行输入3个数，分别为该课程的学分，得分，是否为荣誉课（是为1，否为0）。

输出

输出 n 行，每行两个数，分别为每个学生的学号和其总成绩。（用空格隔开，输出顺序与输入顺序一致，成绩保留到小数点后四位）

输入样例

```
1 3
2 1 3
3 1 99 0
4 4 92 1
5 2 83 1
6 2 2
7 2 100 1
8 3 75 0
9 3 3
10 9 82 1
11 3 90 0
12 5 85 1
```

输出样例


```
1 1 84.9353
2 2 79.5341
3 3 91.5435
```

数据范围

1. $0 < n \leq 50$
2. $0 < m \leq 30$
3. 学号为最多8位的正整数
4. 每门课的得分为整数且 $0 \leq score \leq 100$
5. 每门课的学分为整数或小数部分为.5的浮点数且 $0 < credit \leq 10$
6. 每名同学至少修了一门荣誉课

HINT

为什么会有这么复杂的排名系统？我也想知道

Author: Arthas

开心的华胖胖

题目描述

现在有 n 个活动，每个活动有自己的快乐值 h_i 。请你从这些活动中帮华胖胖找到一段连续的活动，使其快乐值之和最大，并输出该快乐值之和。

输入

共两行。

第一行一个正整数 n ，表示有 n 个活动。

第二行 n 个整数，表示每个活动的快乐值 h_i 。

输出

一行，一个整数，表示连续活动快乐值之和的最大值。

输入样例

```
1 9
2 -2 1 -3 4 -1 2 1 -5 4
```

输出样例

```
1 6
```

样例解释

$[4, -1, 2, 1]$ 这一段连续活动的快乐值之和最大为6，故输出6

数据范围

$$1 \leq n \leq 100000$$

$$-10^4 \leq h_i \leq 10^4$$

H zhn の 字符串

题目描述

zhn在疯狂星期四中花光了自己所有的钱（QAQ），于是他和yh决定去通过探险来获得更多的Money，现在有这样一道带有密码的门挡在了他们面前。

门上有一串由'0'-'9'组成的、长度为 n 的字符串，并且有一个数字 k ，门的密码是：你在这串数字中添加 k 个'+'号（**添加后表达式合法**），然后所有拆分所得的**所有**合法表达式之和。（具体看输入样例）由于这个数字很大，你只需要输出他对 $10^9 + 7$ 取模的答案即可。

众所周知，yh是数学爷爷，他很快就解决了这道题，你们能帮帮zhn这个baka来解决一下这个问题吗？

输入

第一行包含两个整数， n 和 k ($0 \leq k < n \leq 10^5$)

第二行包含一个由 n 个数字组成的字符串。

输出

输出对 $10^9 + 7$ 取模的答案

输入样例

```
1   3 1
2   234
```

输出样例

```
1   63
```

样例解释: $(2 + 34) + (23 + 4) = 63$

Hint1:

如果你直接暴力的话, 是明显过不去的, 考虑一下如何通过组合数来计算每个数字对总答案的贡献。

Hint2:

逆元基本概念:

逆元的定义: 正整数 a, n , 如果有 $ax \equiv 1 \pmod n$, 则称 x 的最小正整数解为 a 模 n 的逆元。

更直白地解释上面的定义: 在模 n 意义下, 一个数 a 如果有逆元 x , 那么 **除以** a 相当于 **乘以** x 。

一个数有逆元的充分必要条件是: $\gcd(a, n) = 1$ 。通常我们设 a 的逆元为 $Inv(a)$

为什么需要逆元:

当我们要得到 $\frac{a}{b} \% p$ 的值, 如果 b 很大, 会爆掉精度。无法直接求得 $\frac{a}{b}$ 的值时, 我们就要用到乘法逆元。

如何求逆元?

本题为了简化以及减少篇幅, 保证给定的**模数 p 为素数**, 通过 *Fermat* 定理, 可以快速求得 a 的逆元为:

$$Inv(a) = a^{p-2} \% p$$

Hint3:

可爱的zhn为了降低难度, 把大部分代码 (所有的函数) 都写好供你直接使用, 具体如下:

```
1   #include <stdio.h>
2   #define M 1000000007
3   #define N 100005
4   #define ll long long
5   int n,k;
6   /*
7   大概你还需要补充一些变量
8   */
9   ll fac[N], inv_fac[N];
10  ll ksm(ll a, ll b) {
11      ll temp=1;
12      while(b){
13          if(b&1) temp=temp*a%M;
14          a=a*a%M;
```

```

15         b>>=1;
16     }
17     return temp;
18 }
19 ll inv(ll x){
20     return (ksm(x,M-2)%M+M)%M;
21 }
22 void init(){
23     fac[0]=1;
24     for(int i=1;i<=n; i++) fac[i]=fac[i-1]*i%M;
25     for(int i=0;i<=n;i++) inv_fac[i]=inv(fac[i]);
26 }
27 ll C(ll a,ll b){
28     if(a<b) return 0;
29     return fac[a]*inv_fac[b]%M *inv_fac[a-b]%M;
30 }
31 int main(){
32     /*
33         在这里补全你的代码
34         这道水水题，相信你一定可以随便切掉的吧！
35     */
36     return 0;
37 }

```

Hint4:

这里的合法指的是加号位置合法，而00+00这种式子我们也可以把他当作合法的，0000+这种东西就是不合法的

Author: 魔法少女zhn

后记：不要问 x 去哪里了， x 被我俩关小黑屋了。

I 方阵数局简单版

本题为C5预习题，二维数组

题目描述

有一个 $m \times n$ 的方阵，用数字 1, 2, 3, 4 填满，如图所示

2	2	2	3
1	3	4	3
1	4	4	4

在其之上有如下移动规则：

- 若某一方格上的数为 1，则可向上移动一格；
- 若某一方格上的数为 2，则可向右移动一格；
- 若某一方格上的数为 3，则可向下移动一格；
- 若某一方格上的数为 4，则可向左移动一格；

现在，给你一个这样的数字方阵，并给你初始位置，请判断经过 t 次移动后，所在的位置在哪里

输入格式

输入共 $n + 2$ 行；

第一行两个正整数 n, m ，表示方阵的大小；

接下来 n 行，每行 m 个空格分隔的整数，表示方格上的数 $a_{ij}, a_{ij} \in 1, 2, 3, 4$ ；

最后一行三个整数 x, y, t ，表示初始位置的坐标 (x, y) 和移动次数 t ；

规定左上角方格的坐标是 $(1, 1)$ ，往右一个方格的坐标是 $(1, 2)$ ，往下一个方格的坐标是 $(2, 1)$ ；

输出格式

输出一个坐标，形如 (x', y') ，表示经过 t 次移动后所在的位置坐标；

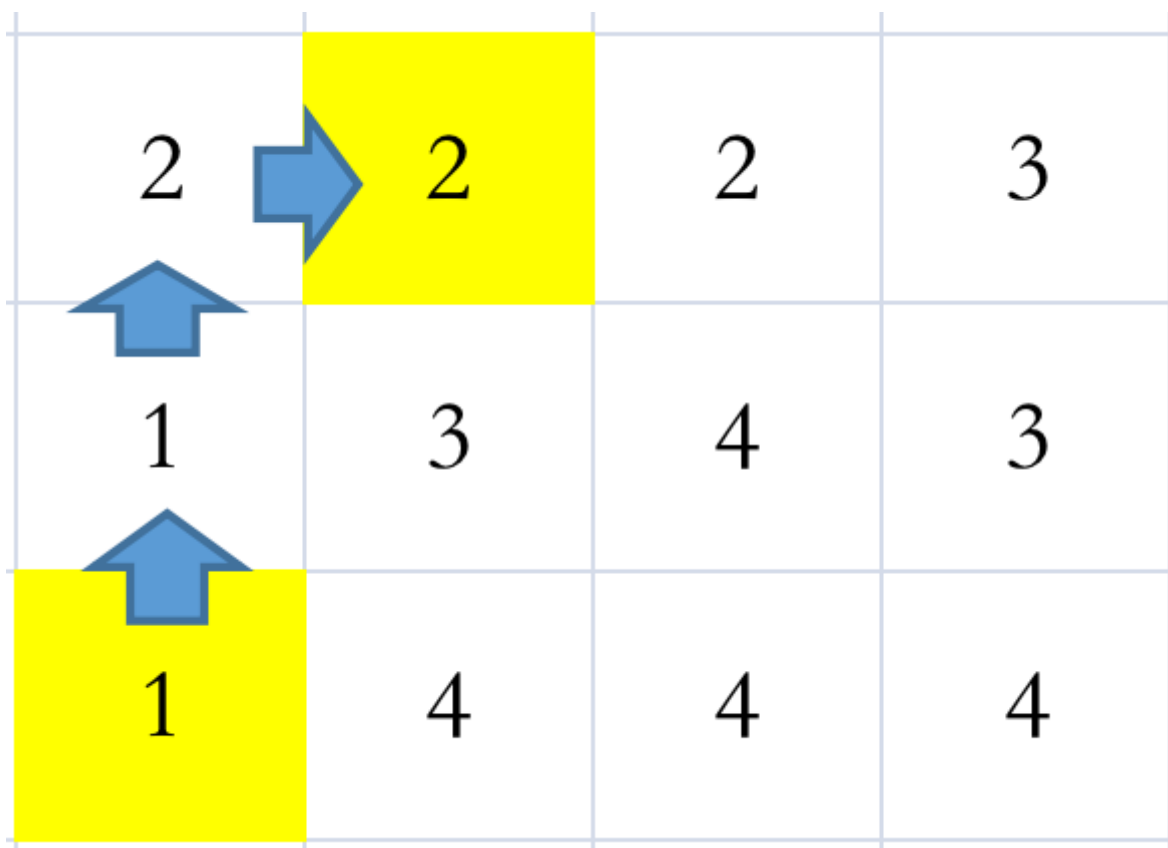
输入样例

```
1  3 4
2  2 2 2 3
3  1 3 4 3
4  1 4 4 4
5  3 1 3
```

输出样例

```
1  (1,2)
```

样例解释



数据范围

保证初始位置在方阵中，保证移动过程中不会超出方阵

$1 \leq n, m \leq 1000$

$0 \leq t \leq 1000$

AUTHOR: 陈博胆

本题为C5预习题，冒泡排序

题目介绍

每次上机之后助教们都要统计上机成绩，可是这周助教被各种作业和ddl缠身，同时还要给同学们答疑，所以拜托了阿水来帮忙做成绩分析。

现在给出 n 个数（每个数可能是整数，也可能是小数），需要从小到大依次输出。

输入格式

输入共 $n+1$ 行；

第一行为一个整数 n ，表示有 n 个数

接下来有 n 行，每行一个数，表示成绩

输出格式

输出共 n 行，每行一个数（保留小数点后五位），按成绩从小到大输出

输入样例

```
1 5
2 3.06
3 1.1
4 4
5 5
6 10
```

输出样例

```
1 1.10000
2 3.06000
3 4.00000
4 5.00000
5 10.00000
```

数据范围

$0 < n < 1001$,所有数据均不会超过 `double` 范围

K Pure Brightness

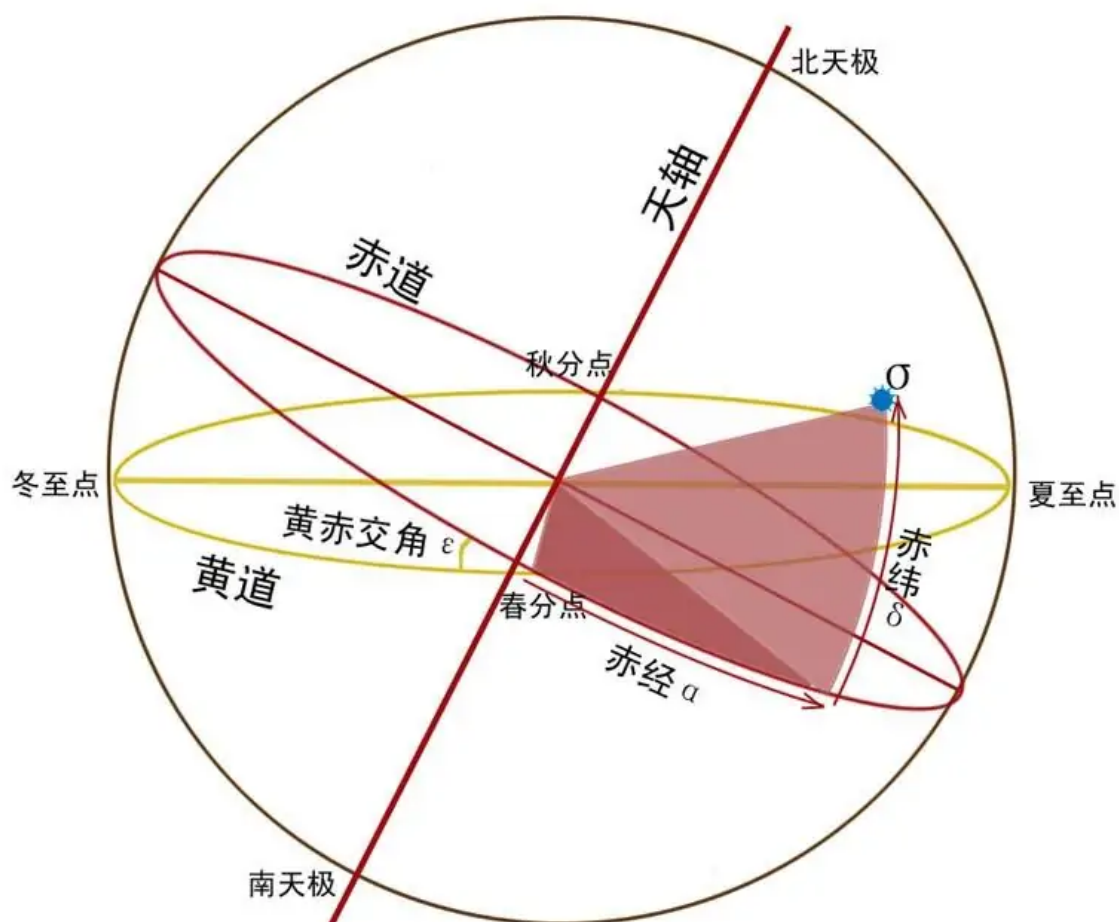
本题为C5预习题

题目描述

引言：农田之神，春天之神。飞过洗翠的天空，带来生命的新芽。试问春神，你从何处来？春神无语，只知寒冬一去春自来。——《古老诗文17》

清明节（Pure Brightness）是二十四节气（24 Solar Terms）之一。不久前第24届冬季奥运会的开幕式，恰好是立春那天，大量出现了中国古典二十四节气的元素。这令呱呱泡蛙回想起了小时候学过的天文学与历法学知识。

天球坐标系有赤道坐标系和黄道坐标系。黄道和赤道的交点为“春分点”和“秋分点”，黄赤交角约为 $\varepsilon = 23^\circ 26'$ 。赤经和黄经以春分点为起始 0° ，绕一周均为 360° 。赤纬和黄纬以赤道和黄道为 0° ，向北为正值，向南为负值。



二十四节气将黄道切割为二十四等分。太阳在黄道上运行一周，运行到某个等分点的位置，所在当天称为节气。

坐标换算的方法，需要借助直角球面三角形的知识。作为一道C语言程序设计的题目，你不需要进行几何学推理。如果真的感兴趣，可以业余时间查资料推理推理。

二十四节气的规范英文表述与太阳黄经对照表，已经由呱呱泡蛙列在Hint。接下来要完成和赤道坐标系的换算，即计算二十四节气当天，太阳的赤经和赤纬。换算的公式也由呱呱泡蛙推理完毕，直接给出：

已知黄道上的黄纬 $\beta = 0^\circ$ ，黄经 λ ，则赤纬 δ 可以直接计算：

$$\delta = \arcsin(\sin 23^\circ 26' \sin \lambda)$$

赤经 α 则是有关系：

$$\tan \alpha = \cos 23^\circ 26' \tan \lambda$$

不仅需要借助反三角 `atan` 函数，还要根据黄经 λ 的范围，以及几何关系，将赤经 α 锁定进相应的 $0^\circ \rightarrow 360^\circ$ 范围内。由图可知，黄经 λ 和赤经 α 总是同时属于 $0^\circ \rightarrow 90^\circ$ 、 $90^\circ \rightarrow 180^\circ$ 、 $180^\circ \rightarrow 270^\circ$ 和 $270^\circ \rightarrow 360^\circ$ 的相同区间。

注意当黄经 λ 为某些特殊值的时候，表达式 $\tan \lambda$ 的值不存在，但是赤经 α 却是存在的。

输入

多行输入。每行是一个可能含有若干空白字符（例如空格）与标点、不区分大小写的字符串。每行字符串长度不超过100。

每行的末尾可能是 `\n` 或者 `E0F`（最后一行）。

输出

对于每行输入的字符串：

去掉其中的空白字符。

如果和二十四节气的某一个英文名匹配，按照下文规定格式，输出当天太阳位置相应的赤经和赤纬，单位用小数形式的角度制（而不是度分秒），保留 3 位小数。

规定格式是，左括号，赤经，逗号，空格，赤纬，右括号，换行符。

如果不能和二十四节气的英文名匹配，输出“This day is not one of the 24 Solar Terms.”

输入样例

```
1   p u   R e b r   i G h T n E s s
2   Frost's Descent
3   FrostsDescent
```

输出样例

```
1 (13.812, 5.988)
2 (207.912, -11.469)
3 This day is not one of the 24 Solar Terms.
```

样例解释

“霜降”的英文少了标点（该标点写在C程序中需要转义），就无法匹配。

Hint

以下翻译均来源于奥运会版，本题按照以下翻译标准进行，其他的翻译版本不予通过。

（四六级热点押题进行时）

太阳 黄经	中 文	English	太阳 黄经	中 文	English	太阳 黄经	中 文	English	太阳 黄经	中 文	English
0°	春 分	Spring Equinox	15°	清 明	Pure Brightness	30°	谷 雨	Grain Rain	45°	立 夏	Beginning of Summer
60°	小 满	Grain Buds	75°	芒 种	Grain in Ear	90°	夏 至	Summer Solstice	105°	小 暑	Minor Heat
120°	大 暑	Major Heat	135°	立 秋	Beginning of Autumn	150°	处 暑	End of Heat	165°	白 露	White Dew
180°	秋 分	Autumn Equinox	195°	寒 露	Cold Dew	210°	霜 降	Frost's Descent	225°	立 冬	Beginning of Winter
240°	小 雪	Minor Snow	255°	大 雪	Major Snow	270°	冬 至	Winter Solstice	285°	小 寒	Minor Cold
300°	大 寒	Major Cold	315°	立 春	Beginning of Spring	330°	雨 水	Rain Water	345°	惊 蛰	Awakening of Insects

- 对于空白字符，可以考虑 `isspace` 函数。对于不区分大小写的情形，可以考虑 `isupper` 和 `tolower` 函数。对比两个字符串是否相同，可以考虑 `strcmp` 函数。相应的用法和头文件可以自行学习。
- C语言中默认使用弧度制。如果程序中要写角度制，需要自行换算成弧度制。输出也要将弧度制换成角度制输出。
- C语言中一律取 π 为 `acos(-1)`，只有这个值是最接近 π 的浮点数。
- 使用 `acos(-1)` 或者 `4*atan(1)` 写出来的 π 是：

```
1 3.14159265358979310000
```

- 采用其他值，例如 `acos(-1.0/2.0)`，`acos(1.0/2.0)`，`asin(1.0/2.0)` 等等，写出来的 π 是：

```
1 3.14159265358979360000
```

- 这就不是最接近 π 的浮点数了。

写在后面

一个夏天的傍晚，茶后无事，我们便海阔天空、东拉西扯地闲聊起来，从高尔夫球俱乐部到黄赤交角变化的原因。——《希腊译员·福尔摩斯探案集》

今天的中国传统农历，完全由天文观测决定。该历法需要解决两个问题：农历月从什么时候起始，什么样的农历月是闰月。

定朔即借助“朔”落在哪一天确定一个农历月的起始。唐代以前的历法都用平朔，自李淳风改进算法的麟德历起，定朔一直沿用至今。

闰月问题的解决，离不开二十四节气。二十四节气分为十二个节令（例如四立日）和十二个中气（例如二分二至日），当一个农历月不含中气的时候，记为农历的闰月。

太阳运行到春分点，是天文学意义的一年起始，这个传统始于西方古希腊天文学。西方古代的天文与占星术是一套体系，占星术以春分点为起始，将黄道等分为十二星宫，这与东方的二十四节气系统不谋而合、完美兼容。



Author: 喜欢看星星看月圆的呱呱泡蛙
