

# 本章主要考模拟法

## 译题 - ride

### 描述

科学家们在研究彗星后惊讶地发现，在每一个彗星后面都有一个不明飞行物 UFO。这些不明飞行物时常来带走来自地球上的一些支持者。不幸地，他们的空间在每次旅行只能带上一群支持者。他们要做的是用一种聪明的方案让某个支持彗星 UFO 的团体都被彗星带走。他们为每个彗星起了一个名字，通过这些名字来决定一个团体是不是特定的彗星带走。那个相配方案的细节是这样的：

所有团体的名字和彗星的名字都以下列各项方式转换成一个数字：这个最后的数字代表名字中所有字母的信息，“A”是 1 和 “Z”是 26。

举例来说，团体 “USACO” 会是  $21*19*1*3*15=17955$  。如果团体的数字  $\text{mod } 47$  等于彗星的数字  $\text{mod } 47$ ，那么你要告诉这个团体：准备好行李，走吧 ！

现在，你要写一个程序来通过团体的名字和彗星的名字来决定一个组是否应该与在那一颗彗星后面的不明飞行物搭配。

写一个程序读入彗星的名字和团体的名字，如果搭配打印 “GO” 否者打印 “STAY”

团体的名字和彗星的名字将会是没有空格或标点的一串大写字母（不超过 6 个字母），

### 格式

文件名： ride(.pas/.c/.cpp)

输入格式：（输入文件名 ride.in）

第 1 行：

彗星的名字（一个长度为 1 到 6 的字符串）

第 2 行：

团体的名字（一个长度为 1 到 6 的字符串）

输出格式：（输出文件名 ride.out）

只有一行----“STAY”或“GO”.

## 样例

样例 1:

输入:  
COMETQ  
HVNGAT  
输出:  
GO

样例 2:

输入:  
ABSTAR  
USACO  
输出:  
STAY

## 解题报告:

本题按照题意模拟即可.

# 译题 - gift1

## 描述

对于一群要互送礼物的朋友，你要确定每个人送出的礼物比收到的多多少 (and vice versa for those who view gift giving with cynicism)。

在这一个问题中，每个人都准备了一些钱来送礼物，而这些钱将会被平均分给那些将收到他的礼物的人。然而，在任何一群朋友中，有些人将送出较多的礼物 (可能是因为有较多的朋友)，有些人有准备了较多的钱。

给出一群朋友，没有人的名字会长于 14 字符，给出每个人将花在送礼上的钱，和将收到他的礼物的人的列表，请确定每个人收到的比送出的钱多的数目。

## 格式

**PROGRAM NAME:** gift1

**INPUT FORMAT:**

(file gift1.in)

第 1 行: 人数 NP,  $2 \leq NP \leq 10$

第 2 到 NP+1 行: 这 NP 个在组里人的名字 一个名字一行

第 NP+2 到最后:

这里的 NP 段内容是这样组织的:

第一行是将会送出礼物人的名字。

第二行包含二个数字: 第一个是原有的钱的数目 (在 0 到 2000 的范围里), 第二个  $NG_i$  是将收到这个送礼者礼物的人的个数 如果  $NG_i$  是非零的, 在下面  $NG_i$  行列出礼物的接受者的名字, 一个名字一行。

### OUTPUT FORMAT:

(file gift1.out)

输出 NP 行

每行是一个的名字加上空格再加上收到的比送出的钱多的数目。

对于每一个人, 他名字的打印顺序应和他在输入的 2 到 NP+1 行中输入的顺序相同。所有的送礼的钱都是整数。

每个人把相同数目的钱给每位要送礼的朋友, 而且尽可能多给, 不能给出的钱被送礼者自己保留。

### IMPORTANT NOTE:

测试系统是 Linux 符合标准的 Unix 的协定。用 '\n' 作为行的结束。这和 Windows 系统用 '\n' 和 '\r' 作为行的结束是不同的。 你的程序不要被这困住了。

## SAMPLE INPUT

```
5
dave
laura
owen
vick
amr
dave
200 3
laura
owen
```

```
vick
owen
500 1
dave
amr
150 2
vick
owen
laura
0 2
amr
vick
vick
0 0
```

## SAMPLE OUTPUT

```
dave 302
laura 66
owen -359
vick 141
amr -150
```

## 解题报告:

同样应用模拟法,下面列出主程序,只要记录收入与支出即可.

```
for i:=1 to n do
begin
  readln(x);
  for j:=1 to n do
    if a[j].zh=x then begin readln(a[j].mon,a[j].s);break;end;
    for k:=1 to a[j].s do
      begin
        temp:=trunc(a[j].mon div a[j].s);
        readln(x);
        for q:=1 to n do
          if a[q].zh=x then a[q].h:=temp+a[q].h;
        {遍历名字列表,将钱送出}
        a[j].h:=a[j].h-temp;{记录支出}
      end;
    end;
  end;
```

# . 译题 - friday

## 描述

13 号又是星期五是一个不寻常的日子吗?13 号在星期五比在其他日少吗?为了回答这个问题, 写一个程序来计算在  $n$  年里 13 日落在星期一, 星期二..... 星期日的次数. 这个测试从 1900 年 1 月 1 日到  $1900+n-1$  年 12 月 31 日.  $n$  是一个非负数且不大于 400.

这里有一些你要知道的:

1900 年 1 月 1 日是星期一. 4, 6, 11 和 9 月有 30 天. 其他月份除了 2 月都有 31 天. 闰年 2 月有 29 天, 平年 2 月有 28 天. 年份可以被 4 整除的为闰年 (1992=4\*498 所以 1992 年是闰年, 但是 1990 年不是闰年) 以上规则不适用于世纪年. 可以被 400 整除的世纪年为闰年, 否则为平年. 所以, 1700, 1800, 1900 和 2100 年是平年, 而 2000 年是闰年. 请不要预先算好数据!

## 格式

PROGRAM NAME: friday

INPUT FORMAT:

(file friday.in)

一个整数  $n$ .

OUTPUT FORMAT:

(file friday.out)

七个在一行且相分开的整数, 它们代表 13 日是星期六, 星期日, 星期一... 星期五的次数.

## SAMPLE INPUT

20

## SAMPLE OUTPUT

36 33 34 33 35 35 34

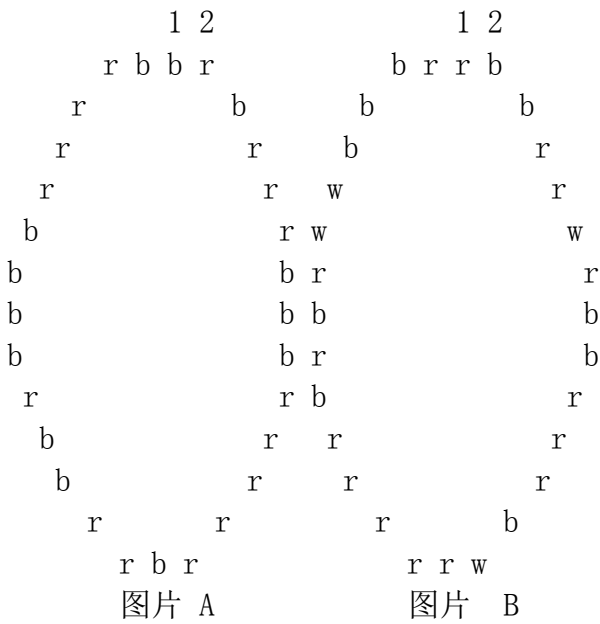
解题报告:

这道题要注意周期性，定义 1900.1.1 是周一，而并不是每年的 1.1 都是周一，方法是直接计算，（13+31）也就是说二月是第一个被计算的，这样最后就多算了一个月，少算了头，补上即可。  
而（0+13）从一月计算则需考虑周期性。

译题 - beads

描述

你有一条由 N 个红色的，白色的，或蓝色的珠子组成的项链 (3<=N<=350)，珠子是随意安排的。 这里是 n=29 的二个例子：



r 代表 红色的珠子  
b 代表 蓝色的珠子  
w 代表 白色的珠子

第一和第二个珠子在图片中已经被作记号。

图片 A 中的项链可以用下面的字符串表示：

brbrrrrbbrrrrrrbrrbbrbbbbrrrrrb .

假如你要在某点打破项链, 展开成一条直线, 然后从一端开始收集同颜色的珠子直到你遇到一个不同的颜色珠子, 在另一端做同样的事。(颜色可能与在这之前收集的不同) 确定应该在哪里打破项链来收集到最大数目的子。

**Example** 举例来说, 在图片 A 中的项链, 可以收集到 8 个珠子, 在珠子 9 和珠子 10 或珠子 24 和珠子 25 之间打断项链。 在一些项链中, 包括白色的珠子如图片 B 所示。当收集珠子的时候, 一个被遇到的白色珠子可以被当做红色也可以被当做蓝色。 表现项链的字符串将会包括三符号 r , b 和 w 。 写一个程序来确定从一条被供应的项链最大可以被收集珠子数目。

## 格式

**PROGRAM NAME:** beads

**INPUT FORMAT:**

(file beads.in)

第 1 行: N, 珠子的数目

第 2 行: 一串度为 N 的字符串, 每个字符是 r , b 或 w。

**OUTPUT FORMAT:**

(file beads.out)

单独的一行包含从被供应的项链可以被收集的珠子数目的最大值。

## SAMPLE INPUT

29

wwwbbrwrbrbrrrbrrbrwrwwrbwrwrrb

## SAMPLE OUTPUT

11

## 输出解释

考虑两段项链（能组成一圈的）。11 个珠子已经标注了。

wwwbbrwrbrbrrrbrrbrwrwwrbwrwrrb wwwbbrwrbrbrrrbrrbrwrwwrbwrwrrb

\*\*\*\*\*

## 解题报告：

一：这道题用标准的搜索是  $O(n^2)$  的，可以用类似动态规划的方法优化到  $O(n)$ 。

用数组  $b_l, b_r, r_l, r_r$  分别记录在项链  $i$  处向左向右收集的蓝色红色珠子数。

项链是环形的，但我们只要把两个同样的项链放在一块，就把它转换成线性的了。

我们只要求出  $b_l, b_r, r_l, r_r$ ，那么结果就是

$\max(\max(b_l[i], r_l[i]) + \max(b_r[i+1], r_r[i+1]))$  ( $0 \leq i \leq 2*n-1$ )。

我们以求  $b_l, r_l$  为例：

初始时  $b_l[0]=r_l[0]=0$

我们从左往右计算

如果  $necklace[i]='r'$ ， $r_l[i]=r_l[i-1]+1, b_l[i]=0$ ；

如果  $necklace[i]='b'$ ， $b_l[i]=b_l[i-1]+1, r_l[i]=0$ ；

如果  $necklace[i]='w'$ ， $b_l[i]=b_l[i-1]+1, r_l[i]=r_l[i-1]+1$ 。

同理可以求出  $b_r, r_r$ 。

二：可以直接模拟。