

2015 株洲二中 7 月模拟赛 1

一、题目概况

中文题目名称	报数 7	塔的最大高度	亚瑟王的游戏	猴子
子目录名	seven	tower	camelot	monkey
可执行文件名	seven.exe	tower.exe	camelot.exe	monkey.exe
每个测试点时限	1000 毫秒	1000 毫秒	1000 毫秒	1000 毫秒
测试点数目	10	10	20	10
每个测试点分值	10	10	5	10
附加样例文件	无	无	无	无
结果比较方式	全文比较			
题目类型	传统	传统	传统	传统
运行内存上限	128 兆字节	128 兆字节	128 兆字节	128 兆字节

二、提交源程序文件名

对于 C 语言	seven.c	tower.c	camelot.c	monkey.c
对于 C++语言	seven.cpp	tower.cpp	camelot.cpp	monkey.cpp

三、编译命令

对于 C 语言	gcc -Wall -std=c99 -O2 -DOJ -o foo src.c -lm
对于 C++语言	g++ -Wall -std=c++11 -O2 -DOJ -o foo src.cpp -lm

注意事项:

- 1、比赛使用标准输入输出。
- 2、文件名必须使用英文小写。
- 3、C/C++中函数 main 的返回值类型必须是 int，程序正常结束时返回值必须是 0。
- 4、评测时采用的机器配置为：Intel Core i5-3470 3.2 GHz × 2 处理器，8GB 内存。上述时限以此配置为准。
- 5、特别提醒：评测在 CentOS 6.6 x86_64 操作系统上进行，各语言的编译器版本以其为准。GCC 版本目前为 5.1.0。

1、报数 7

【问题描述】

1337 个人围成一个圈，编号依次为 1,2,3...1337。从 1 号人开始报数，初始的报数方向是 1, 2, 3...。如果某个人报的数是 7 的倍数或者数字中含有 7，那么报数的方向就反向一次。问报数字 x 的是那个人的编号。

【输入】

只有一行，正整数 x 。

【输出】

最终报数字 x 的那个人的编号。

【输入输出样例】

Input	Output
1000	1311

样例说明：

初始的 15 个报数情况是：(人的编号->报的数字)

1->1, 2->2, 3->3, 4->4, 5->5, 6->6, 7->7, 6->8, 5->9, 4->10, 3->11, 2->12, 1->13, 1337->14, 1->15

【数据说明】

对于 30%的数据， $x \leq 10^6$;

对于 100%的数据， $x \leq 10^9$;

2、塔的最大高度

【问题描述】

给出 N 个木块，告诉你每个木块的高度，你要用这些木块搭出两座高度相同的塔，一个塔的高度为搭建它的木块的高度之和，并且一个塔至少要用一个木块。木块只能用一次，也可以不用。问在这两座塔高度相同的条件下，能够搭出的塔的最大高度是多少？

【输入】

第一行一个整数 N ($N \leq 50$) ,表示木块的个数；

第二行 N 个整数，表示 N 个木块的高度。

每个木块的高度范围 $[1,500000]$,所有木块的高度总和 ≤ 500000 。

【输出】

一个数，表示能搭建的塔的最高高度，若不能搭建两座相同的塔，则输出-1.

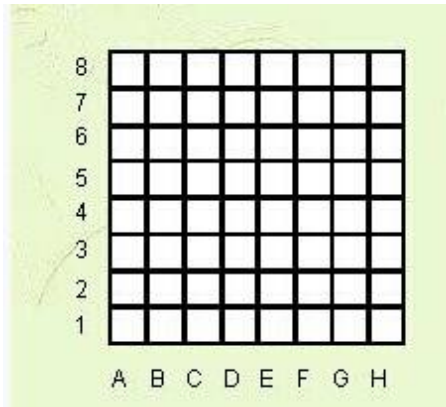
【输入输出样例】

Input	Output
3 2 3 5	5

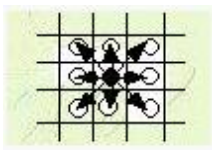
3、亚瑟王的游戏

【问题描述】

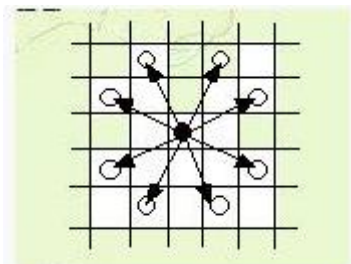
很久以前，亚瑟王和他的骑士习惯每年元旦去庆祝他们的友谊。为了纪念上述事件，我们把这些是看作是一个有一人玩的棋盘游戏。有一个国王和若干个骑士被放置在一个由许多方格组成的棋盘上，没有两个骑士在同一个方格内。这个例子是标准的 8*8 棋盘



国王可以移动到任何一个相邻的方格，从下图中黑子位置到下图中白子位置前提是他不掉出棋盘之外。



一个骑士可以从下图中黑子位置移动到下图白子位置(走“日”字形) 但前提是他不掉出棋盘之外。



在游戏中，玩家可在每个方格上放不止一个棋子，假定方格足够大，任何棋子都不会阻碍到其他棋子正常行动。

玩家的任务就是把所有的棋子移动到同一个方格里——用最小的步数。为了完成这个任务，他必须按照上面所说的规则去移动棋子。另外，玩家可以选择一个骑士跟国王从他们两个相遇的那个点开始一起行动，这时他们按照骑士的行动规则行动，其他的单独骑士则自己一直走到集中点。骑士和国王一起走的时候，只算一个人走的步数。

写一个程序去计算他们集中在一起的最小步数，而且玩家必须自己找出这个集中点。当然，这些棋子可以在棋盘的 anywhere 集合。

【输入】

第一行： 两个用空格隔开的整数：R,C 分别为棋盘行和列的长。不超过 26 列，40 行。

第二行..结尾： 输入包含了一些有空格隔开的字母/数字对，一行有一个或以上。第一对为国王的位置，接下来是骑士的位置。可能没有骑士，也可能整个棋盘都是骑士。行从 1 开始，

列从大写字母 A 开始。

【输出】

单独一行表示棋子集中在一个方格的最小步数。

【输入输出样例】

Input	Output
8 8 D 4 A 3 A 8 H 1 H 8 国王位置在 D4。一共有四个骑士，位置分别是 A3,A8,H1 和 H8。	10

【样例解释】

他们集中在 B5。

骑士 1: A3 - B5 (1 步)

骑士 2: A8 - C7 - B5 (2 步)

骑士 3: H1 - G3 - F5 - D4 (此时国王开始与这个骑士一起走) - B5 (4 步)

骑士 4: H8 - F7 - D6 - B5 (3 步)

$1 + 2 + 4 + 3 = 10$ 步

4、猴子

【问题描述】

有 N 只猴子，第一只尾巴挂在树上，剩下的 $N-1$ 只，要么被其他的猴子抓住，要么抓住了其他的猴子，要么两者均有。当然一只猴子最多抓住两只另外的猴子，猴子只有两只手。现在给出这 N 只猴子抓与被抓的信息，并且在某个时刻可能某只猴子会放掉它的左手或右手的猴子，导致某些猴子落到地上。求每只猴子的落地时间。

【输入】

第一行两个数 N, M ，表示有 N 只猴子，并且总时间为 $M-1$ 。

接下来 N 行，描述了每只猴子的信息，每行两个数，分别表示这只猴子左手和右手抓的猴子的编号，如果是 -1 表示该猴子那只手没有抓其他的猴子。

再接下来 M 行，按时间顺序给出了一些猴子放手的信息，第 $1+N+i$ 行表示第 $i-1$ 时刻某只猴子的放手信息，信息以两个数给出，前者表示放手的猴子的编号，后者表示其放手的是哪只手，1 表示左手，2 表示右手。

【输出】

输出 N 行，第 i 行表示第 i 只猴子掉落时刻，若猴子到 $M-1$ 时刻以后还没有掉落，就输出 -1。

【输入输出样例】

Input	Output
3 2	-1
-1 3	1
3 -1	1
1 2	
1 2	
3 1	

【数据说明】

30%的数据满足： $N \leq 1000, M \leq 1000$;

100%的数据满足： $1 \leq N \leq 200000, 1 \leq M \leq 400000$ 。