

全国信息学奥林匹克联赛（CSP-J2022）复赛_仿真

普及组

（请选手务必仔细阅读本页内容）

一. 题目概况

中文题目名称	Astro	Hrpa	Why Did the Cow Cross the Road II-P	Dugovi
英文题目与子目录名	astro	hrpa	nocross	dugovi
可执行文件名	astro	hrpa	nocross	dugovi
输入文件名	astro.in	hrpa.in	nocross.in	dugovi.in
输出文件名	astro.out	hrpa.out	nocross.out	dugovi.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	20	10	15	10
每个测试点分值	5	10	1-5(6), 6-15(7)	10
附加样例文件	有	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统

二. 提交源程序文件名

对于 pascal 语言	astro.pas	hrpa.pas	nocross.pas	dugovi.pas
对于 c 语言	astro.c	hrpa.c	nocross.c	dugovi.c
对于 c++ 语言	astro.cpp	hrpa.cpp	nocross.cpp	dugovi.cpp

三. 编译命令（不包含任何优化开关）

对于 pascal 语言	fpc astro.pas	fpc hrpa.pas	fpc nocross.pas	fpc dugovi.pas
对于 c 语言	gcc -o astro astro.c -lm	gcc -o hrpa hrpa.c -lm	gcc -o nocross nocross.c -lm	gcc -o dugovi dugovi.c -lm
对于 c++ 语言	g++ -o astro astro.cpp -lm	g++ -o hrpa hrpa.cpp -lm	g++ -o nocross nocross.cpp -lm	g++ -o dugovi dugovi.cpp -lm

四. 运行内存限制

内存上限	128M	128M	256M	128M
------	------	------	------	------

注意事项：

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
- 3、C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
- 4、统一评测时采用的机器配置为：CPU Intel(R) Celeron(R) G3900 @ 2.8GHz，内存 4G，上述时限以此配置为准。
- 5、特别提醒：评测在 **Windows** 下进行，评测软件为 **CCR Plus**。

Astro

(astro.cpp/c/pas)

【题目描述】

Mirko 非常喜欢上天文课，他正在观察两颗不寻常的恒星。他注意到这两颗恒星会不定期地闪现。他进一步注意到，每颗恒星的闪现间隔时间是一样的。现在他想知道，在哪一天，两颗恒星第一次在同一分钟闪现？

例如，如果第一颗恒星今天（星期六）在 2:20 闪现，第二颗恒星在 13:00 闪现。第一颗恒星每隔 05:50（每 5 小时 50 分钟）闪现一次，第二颗恒星每隔 01:00（每小时）闪现一次。第一颗恒星将在 08:10 和 14:00 再次闪现，第二颗恒星在 14:00 闪现。因此这两颗恒星将在今天的 14:00 同时闪现。

注意：时间 00:00（午夜）是新一天的第一分钟。

【输入格式】

输入共四行，都是时间格式 HH:MM（小时:分钟）。四个时间分别是：第一颗恒星的闪光时间，第二颗恒星的闪光时间，第一颗恒星的闪现间隔时间，第二颗恒星的闪现间隔时间。前两个时间会有所不同，并且都是同一天——周六。闪现间隔时间不会是 00:00。

【输出格式】

如果两颗恒星永远不会在同一分钟闪现，则输出“Never”。否则，在第一行输出第一次同时闪现的星期。提醒星期的正确拼写为：“Sunday”，“Monday”，“Tuesday”，“Wednesday”，“Thursday”，“Friday”，“Saturday”。（注意单词首字母大写）；第二行中，输出第一次同时闪现的时间，格式为 HH:MM（如果 HH<10，则注意前导零）。

【输入样例 1】

02:20
13:00
05:50
01:00

【输出样例 1】

Saturday
14:00

【输入样例 2】

02:20
23:28
00:40
23:50

【输出样例 2】

Never

【输入样例 3】

23:19
10:19
02:42
09:11

【输出样例 3】

Thursday

00:31

【数据规模】

对于 40%的数据：两颗恒星的闪现时间均为整小时 (MM 都为 00)；

对于 100%的数据： $0 \leq HH \leq 23$ ； $00 \leq MM \leq 59$ ；

Hrpa

(hrpa.cpp/c/pas)**【题目描述】**

Mirko 和 Slavko 最喜欢在数学游戏中比赛。这次他们拿到一堆鹅卵石，并遵循以下规则：

1. Mirko 是第一个玩，然后是 Slavko，以此类推。
2. Mirko 在第一次操作时从堆中取出任意数量的鹅卵石（1 到 N 之间的数，包括 1 和 N）。
3. 在接下来的每个回合中，当前轮到的玩家至少取 1 个鹅卵石，最多能取前一回合中玩家取的鹅卵石数量的两倍。当然，人们不能取得比堆中剩余数量多的鹅卵石数量。
4. 拿到最后一颗鹅卵石的玩家就是赢家。

Mirko 和 Slavko 都是厉害的玩家（如果一名玩家有可能赢，那么该玩家绝对不会错失良机）。我们想要知道 Mirko 在第一步取的最少数量的鹅卵石，且保证他能赢得比赛。

【输入格式】

输入共一行包含一个正整数 N，表示开始时这一堆鹅卵石的数量。

【输出格式】

输出共一行一个整数，Mirko 在第一回合中需要取走的最少数量的鹅卵石。

【输入样例 1】

4

【输出样例 1】

1

【输入样例 2】

7

【输出样例 2】

2

【输入样例 3】

8

【输出样例 3】

8

【样例 1 说明】Mirko 第一回合中有四种可能的选择：他可以从堆中取走 1、2、3 或 4 颗鹅卵石。如果他取走 4 颗鹅卵石，他必定会赢，但这不是最佳解决方案。我们需要检查其余的情况。如果 Mirko 只取走 1 颗，Slavko 就剩下 3 颗，但他最多可以取 2 颗。Slavko 不能取走所有的鹅卵石，但 Mirko 能够在

下一回合中取回所有剩余的鹅卵石赢得比赛。因此，我们得出结论，1 是该样例最佳的解决方案。

【数据规模】

对于 10%的数据： $2 \leq N \leq 500$ ；

对于 30%的数据： $2 \leq N \leq 5,000$ ；

对于 40%的数据： $2 \leq N \leq 200,000$ ；

对于 100%的数据： $2 \leq N \leq 10^{15}$ ；

Why Did the Cow Cross the Road II-P

(nocross.cpp/c/pas)

【题目描述】

Farmer John 饲养 N 头奶牛，编号为 $1 \dots N$ 。为了牛儿们能产出高品质的牛奶，FJ 可没少花心思，通过一段时间的观察，FJ 认为奶牛是否能产出高品质的牛奶与奶牛在哪些牧区吃草有直接关系。编号为 A 的奶牛在编号为 B 的牧区吃草时，如果 $|A - B| \leq 4$ ，则该头奶牛是快乐的，FJ 也会收获该头奶牛所产出的高品质的牛奶；否则该头奶牛是不快乐的，并且产出的是劣质牛奶，没法食用。

一条漫长的道路穿过 FJ 的农场，刚好将牛棚和牧区隔离开。在道路的一侧有 N 个牛棚（第 i 个牛棚里面的奶牛的编号为 A_i ），道路的另一侧刚好有 N 个牧区（第 i 个牧区的编号为 B_i ）；为了帮助他的奶牛安全的过马路，FJ 想要在马路上画人行横道。每条人行横道都应该将道路一侧的一个牛棚里的奶牛连接到另一侧的一个牧区。

为了安全起见，FJ 不允许一个牛棚连接多个牧区或者多个牛棚连接同一个牧区，更不允许人行横道有交叉的情况。因为这些奶牛在过人行横道的时候会横冲直撞，FJ 可不希望看到他的牛儿们受到伤害。作为产奶大户的 FJ 自然希望更多的奶牛开心快乐，这样他就可以收获更多高品质的牛奶，从而获得更高的利益。

请帮助 FJ 确定他可以在道路上绘制的人行横道的最大数量，使得任意两个人行横道都不相交。

【输入格式】

第 1 行输入包含一个整数 N ；

接下来的 N 行描述了牛棚的信息，整数 A_i 为第 i 个牛棚里面的奶牛的编号；（ A_i 各不相同）

再接下来的 N 行描述了牧区的编号，整数 B_i 为第 i 个牧区的编号；（ B_i 各不相同）

【输出格式】

输出共一行一个整数，即 FJ 在马路上可以画出的最多的不相交的人行横道的数量。

【输入样例】

```
6
1
2
3
4
5
6
6
5
4
3
2
1
```

【输出样例】

5

【数据规模】对于 20%的数据： $1 \leq N \leq 1,000$ ；对于 40%的数据： $1 \leq N \leq 10,000$ ；对于 80%的数据： $1 \leq N \leq 100,000$ ；对于 100%的数据： $1 \leq N \leq 300,000$ ； $1 \leq A_i \leq N$ ； $1 \leq B_i \leq N$ ；

Dugovi

(dugovi.cpp/c/pas)**【题目描述】**

在一个名叫 Kriz 的小镇住着 N 位居民。他们每个人都从其他居民那里借了一些钱。现在是时候偿还所有债务了。但问题是每个人都花光了他们所有的钱！

小镇的镇长决定解决这个问题。他决定拨一笔钱给少数人，以便他们偿还债务。当一些人拿回他们的钱后，就开始了连锁反应。例如：A 从政府获得钱，A 向 B 偿还债务。然后 B 向 C 偿还债务。如果 B 没有足够的钱偿还债务，他就等到他得到足够的钱再偿还。如果他们偿还了债务后还有多余的钱剩下，他们就会将余钱存起来。

举个例子：镇上两个居民，都欠彼此 100 美元，那么政府给他们中的一人 100 美元，这样他们就可以将债务还给另一个。

你的任务是计算政府须向部分居民提供的最低总金额，以便所有人能偿还所有的债务。

【输入格式】

第 1 行输入包含一个整数 N ，表示 Kriz 镇上的居民人数。用 1 到 N 的整数对他们编号。

接下来的 N 行每行包含两个用空格分隔的整数。其中第 i 行中的第一个整数 A_i ($A_i \neq i$)，表示编号为 i 的人需向编号为 A_i 的人偿还债务，第二个整数 B_i 表示偿还的金额。

【输出格式】

输出共一行一个整数，包含政府必须向其居民提供的最低总金额，以便偿还所有债务。

【输入样例 1】

4

2 100

1 100

4 70

3 70

【输出样例 1】

170

【输入样例 2】

3

2 120

3 50

2 80

【输出样例 2】

150

【输入样例 3】

5

3 30

3 20

4 100

5 40

3 60

【输出样例 3】

110

【数据规模】对于 30%的数据： $2 \leq N \leq 300$ ；对于 40%的数据： $2 \leq N \leq 5,000$ ；对于 100%的数据： $2 \leq N \leq 200,000$ ； $1 \leq A_i \leq N$ ； $1 \leq B_i \leq 10,000$ ；