

2018 夏 • 西北工业大学“字节跳动杯”程序设计竞赛

目录

A: 董与黑的碰撞-开端2

B: 董与黑的碰撞-资本.....3

C: 董与黑的碰撞-侵入.....4

D: 董与黑的碰撞-假想6

E: 董与黑的碰撞-点兵8

F: 董与黑的碰撞-伪装 10

G: 董与黑的碰撞-分身 12

H: 董与黑的碰撞-陷阵 13

I: 董与黑的碰撞-决战 15

J: 董与黑的碰撞-彩蛋..... 17

Problem A: 董与黑的碰撞-开端

时间限制: 1000MS

描述:

公元 8102 年, 一个科技高度发达, 文明不断前进的时代, 和平与发展已经成为全宇宙不可更改的趋势。而一款叫做 ACM 的游戏, 由于奥妙无穷, 包含万千, 一度成为了衡量一个星球发达与否的重要指标。此时, 全宇宙在 ACM 上造诣最高的两个星球, 莫过于地球和遥远的 дГНФТНГУY 星球。两星球本可以通过合作的方式, 共同推进全宇宙的发展, 但 дГНФТНГУY 星球上一股强大的黑暗势力突然崛起, 公然无视宇宙条约, 企图对地球发动多方面的制裁。黑暗势力领袖火霸, 尽管掌握着无限宝石的强大能量, 却整天痴迷于 ACM 这款游戏, 一直想要和地球进行一次较量。为了打赢战斗, 证明实力, 地球上最负盛名的 ACM 国际核心玩家联盟 (ACM international central player coalition, 简称 ACM-ICPC), 集结了来自全世界的 ACM 高手, 组建了“复兴者联盟”, 号召全地球为人类的伟大复兴事业而不懈奋斗。谁来担任联盟的盟主, 一度成为了全世界都在议论的话题。ACM 国际核心玩家联盟由于忙着准备战斗, 所以请你来帮忙编写一个评估程序, 通过综合评估所有人的能力, 从而选出盟主。这个评估程序的评估方式是这样的: 如果输入是“dongxiansen”, 则输出“yes”, 否则输出“no”。

输入:

首先输入一个正整数 T, 表示测试数据的组数。

对于每组数据, 输入一个字符串 s, 字符串中没有空格或制表符, 字符串之间用换行符分隔。

数据范围:

$1 \leq T \leq 10$

$1 \leq \text{strlen}(s) \leq 100$

输出:

对于每组测试数据, 输出一个字符串, 表示该组测试数据的答案, 每组输出占一行。

样例输入:

```
2
dongxiansen
npusjx
```

样例输出:

```
yes
no
```

Problem B: 董与黑的碰撞-资本

时间限制: 2000MS

描述:

众所周知, 备战是需要花费大量金钱的, 因此董先森决定用炒股的方式获得更多的资金。

首先, 董先森看中了一支股票, 并通过他强大的脑力预测出了这支股票接下来 n 天的走势, 在第 i 天, 这支股票的价格将是每股 s_i 。董先森拥有两次买卖机会, 但他可以选择只买卖一次或零次。如果董先森买卖两次, 则他必须将第一次所买入的全部卖出之后, 才能进行第二次买入, 也就是说, 董先森必须要在第 a 天买入, 在第 b 天卖出, 再在第 c 天买入, 再在第 d 天卖出, 其中 $1 \leq a < b < c < d \leq n$ 。假设董先森拥有的初始资金是非零的, 并且股票买卖的股数可以是分数, 请问他所能获得的最大收益率是多少?

输入:

首先输入一个正整数 T , 表示测试数据的组数。

对于每组数据, 第 1 行输入一个正整数 n , 第 2 行输入 n 个正整数 s_i 。

数据范围:

$$1 \leq T \leq 10$$

$$4 \leq n \leq 100000$$

$$1 \leq s_i \leq 100000$$

输出:

对于每组测试数据, 输出答案的百分比形式, 并保留两位小数, 每组输出占一行。

样例输入:

```
2
4
2 3 3 3
6
4 7 5 8 6 9
```

样例输出:

```
50.00%
215.00%
```

Problem C: 董与黑的碰撞-侵入

时间限制: 1000MS

描述:

大战将至, 联盟的机器却相继崩溃, 主管数据库的轩神经过反复检查, 发现这是黑暗势力远程侵入了系统, 篡改了程序, 导致计算机中出现了一种叫做死锁的故障。

死锁的原理是这样的:

考虑这样一种情况: 在数据库中, 多个进程会并发执行, 当一个进程希望访问一个数据时, 会先给这个数据上锁, 直到这个进程结束 (提交修改), 这个锁才会被解开。这样, 其他访问该数据库的进程需要访问该数据时, 需要等待锁解开时才可进行访问。例如 T1 进程先访问了 A 数据, T2 进程在之后想要访问 A 数据, 则需要等待 T1 进程提交修改。我们称这为一次等待, 记做 T2 W T1。

由于给数据加锁这一操作在访问数据前一刻进行, 可能会出现这种情况: T1 访问数据 A, T2 访问数据 B, T1 希望访问数据 B, T2 希望访问数据 A, 此时 B 被 T2 上锁, A 被 T1 上锁, 两个进程均无法结束, 两个进程均进入等待, 死锁发生。这种情况可以表示为 T2 W T1, T1 W T2。同样, 在多个进程并发执行时, 若出现冲突出现闭环 (eg. T1 W T2, T2 W T3, T3 W T1), 也会出现死锁。

由于轩神忙于准备战斗, 所以他请你帮忙解决死锁的问题。现在告诉你一台机器中的 n 个进程 (编号 1,2,3...n) 与 m 个冲突信息, 请你判断这台机器是否存在死锁。

输入:

首先输入一个正整数 T, 表示测试数据的组数。

对于每组数据, 第一行输入两个数 n,m, 接下来 m 行, 每行输入两个整数 a,b 表示(a W b), 保证一组数据中相同的(a,b)不会出现两次

数据范围:

$T \leq 500$

$n \leq 1000$

$m \leq 2000$

$1 \leq a, b \leq n$, 且 $a \neq b$

输出:

对于每组测试数据, 若不存在死锁, 则输出一行 "No Deadlock"; 若存在死锁, 则输出一行 "Deadlock Occurs".

样例输入:

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

样例输出：

No Deadlock
Deadlock Occurs
Deadlock Occurs

Problem D: 董与黑的碰撞-假想

时间限制: 2000MS

描述:

为了能够打赢和黑暗势力的战斗, 董先森整天埋头苦思如何应对。这时候一个叫 npusjx 的谋士提出了一个叫“自动 AC 机”的假想, 并且声称它可以完美地将所有题目 AC。

“自动 AC 机”的主要原理是这样的:

①首先, 采用长寿村的神秘方法, 获取评测系统所有输入数据 (方法过于玄妙, 在此不作详述)。假设共有 n 组数据。

②知道了输入数据之后, 仍然想不到解法, 但却可以将每组数据的答案缩小到某个范围内。假设第 i 组数据的答案缩小到了 $[a_i, b_i]$ 这个范围, 并且范围中每个数是答案的概率是等同的。
(这里认为 a_i, b_i 和答案都是整数)

③接下来就是“自动 AC 机”的核心代码:

```
for(i = 1; i <= n; i++) {  
    printf("%d\n", a[i] + k % (b[i] - a[i] + 1));  
}
```

其中 k 按 $0, 1, 2, \dots$ 的顺序, 枚举所有自然数。对于每个 k , 都提交一次, 运气好的话就能 AC。

大家对于这一伟大的假想, 全都拍手称赞。只有董先森看出来 npusjx 是在搅局, 因为这个“自动 AC 机”不仅不稳定, 而且将收获巨大的罚时。现在已知每次错误提交都会增加 20 分钟的罚时, 请问这个“自动 AC 机”在 AC 之前, 总共会收到多少罚时?

注意:

①由于总罚时不是确定值, 请输出总罚时的数学期望。

②由于答案可能很大, 且可能是浮点数, 请输出答案整数部分的后 8 位 (包含前缀零)。

③如果存在永远都无法 AC 的可能, 请输出“ongoing” (不包括引号)。

输入:

首先输入一个正整数 T , 表示测试数据的组数。

对于每组数据, 第 1 行输入一个正整数 n , 接下来 n 行, 每行输入 2 个整数 a_i, b_i 。

数据范围:

$$1 \leq T \leq 8$$

$$1 \leq n \leq 1000$$

$$-10^9 \leq a_i \leq b_i \leq 10^9$$

输出:

对于每组测试数据, 输出 8 位数字, 表示该组测试数据的答案, 每组输出占一行。

样例输入：

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

样例输出：

```
00000010
00000020
```

提示：

对于第二组样例，
k=0 时，自动 AC 机输出 1、1，
k=1 时，自动 AC 机输出 1、2，
k=2 时，自动 AC 机输出 1、3，
有 $\frac{1}{3}$ 的概率答案是 1、1，此时直接 AC，
有 $\frac{1}{3}$ 的概率答案是 1、2，此时收到 20 分钟罚时，
有 $\frac{1}{3}$ 的概率答案是 1、3，此时收到 40 分钟罚时，
最终答案为 $(20/3+40/3)=20$ 。

Problem E: 董与黑的碰撞-点兵

时间限制: 1000MS

描述:

大战在即, 董先森为战斗做着最后的准备, 而选择哪些人去战斗, 成了最重要的问题。已知董先森手下共有 n 个人 a_1, a_2, \dots, a_n , 其中 a_i 表示第 i 个人的战力值。为了配合默契, 董先森决定选择其中一组序号连续的人组成一个大队, 并任命这个区间的最后一人为队长。

对于选定的大队 a_L, a_{L+1}, \dots, a_R , 又会选出由序号连续的人组成的小队, 小队中一定要包含大队的队长。如果这个小队中所有人的战力值的异或值为 1, 这个小队就能打出好的配合。

现在有 m 组询问, 每次询问一个选定的大队 a_L, a_{L+1}, \dots, a_R , 请问这个大队中共有多少组能打出好的配合的小队?

输入:

本题只有一组测试数据。

第一行输入两个正整数 n, m , 第二行输入 n 个整数 a_1, a_2, \dots, a_n , 接下来 m 行, 每行输入两个正整数 L, R 。

数据范围:

$1 \leq n \leq 100000$

$1 \leq m \leq 100000$

$0 \leq a_i \leq 1$

$1 \leq L \leq R \leq n$

输出:

对于每次询问, 输出对应的答案, 每组输出占一行。

样例输入:

```
6 2
1 0 1 0 1 1
1 3
2 6
```

样例输出:

```
2
3
```

提示:

对于样例的第一次询问,

$$a_1 \text{ xor } a_2 \text{ xor } a_3 = 0,$$

$$a_2 \text{ xor } a_3 = 1,$$

$$a_3 = 1,$$

所以答案为 2。

对于样例的第二次询问,

$$a_2 \text{ xor } a_3 \text{ xor } a_4 \text{ xor } a_5 \text{ xor } a_6 = 1,$$

$$a_3 \text{ xor } a_4 \text{ xor } a_5 \text{ xor } a_6 = 1,$$

$$a_4 \text{ xor } a_5 \text{ xor } a_6 = 0,$$

$$a_5 \text{ xor } a_6 = 0,$$

$$a_6 = 1,$$

所以答案为 3。

Problem F: 董与黑的碰撞-伪装

时间限制: 1000MS

描述:

在经历了一系列的准备之后, 地球与 дгнфтнгуы 星球的战斗终于打响, 但 дгнфтнгуы 星球的黑暗势力领袖火霸却突然失去了踪影, 原来是火霸使用祖传的易容术将自己成功伪装在地球的联盟成员之中。但这早已在董先森的预料之中, 董先森事先为所有的联盟成员提供了备身, 只要他一声令下, 联盟成员就会立刻变成原来的双倍, 而火霸没有备身, 因此只要寻找谁的数量为奇数, 就能很快的将火霸找出来。

在生成备身之后, 场上共有 n 个人, 由于逐个辨认的效率太慢, 董先森根据每个人的外观, 生成其对应的一串字符 s_i ($i=1,2,\dots,n$), 请你求出火霸对应的那串字符是什么。

当然, 有可能存在成员与成员之间, 甚至是成员和火霸之间外观相似, 导致对应字符串相同的情况, 但这并不影响最终的答案。

输入:

首先输入一个正整数 T , 表示测试数据的组数。

对于每组数据, 第一行输入一个整数 n , 保证 n 为奇数。接下来 n 行, 每行有一个字符串 s_i , 字符串中字符的 ASCII 码范围为 $[33,127]$ (不含有不可见字符)。

数据范围:

$1 \leq T \leq 5$

$1 \leq n \leq 1000000$

$\text{strlen}(s_i) \leq 10$

输出:

对于每组测试数据, 输出一个字符串, 表示该组测试数据的答案, 每组输出占一行。

样例输入:

```
2
3
A^b
2333
A^b
5
ABCDEF~@
ZZZZZ
ZZZZZ
```

ZZZZZ

ABCDEF~@

样例输出：

2333

ZZZZZ

Problem G: 董与黑的碰撞-分身

时间限制: 1000MS

描述:

意图使用易容术进行伪装的火霸被智慧的地球人成功揪出,恼羞成怒的火霸又使出了祖传的分身术,刹那间召唤出了数以万计的分身,有金霸、木霸、水霸、土霸、风霸、雷霸、电霸、天霸、动霸……,而火霸则藏匿其中,很难被分辨出来。

全宇宙都知道,火霸是一个患有严重强迫症的家伙,这就给了地球人两点线索:

第一,火霸及其分身数量的总和 s 一定满足 $s=2^n-1, n \in \mathbb{N}^+$ 。

第二,火霸无法容忍分身们站队混乱,因此火霸及其分身一定会按照字典序排列。

此时此刻,一个经久不衰的算法从所有地球人的口中呼之欲出,那就是二分大法!

二分大法的具体步骤是这样的:首先查找到最中间的家伙,如果是火霸,则直接揪出,如果不是,则可以根据字典序判断火霸是在左半部分还是右半部分,接着重复上述操作。

已知火霸出现在每个位置的概率是等同的,现在告诉你 n 的大小,请你求出最终查找次数的数学期望。

输入:

首先输入一个正整数 T , 表示测试数据的组数。对于每组数据, 输入一个正整数 n 。

数据范围:

$$1 \leq T \leq 100$$

$$1 \leq n \leq 100$$

输出:

对于每组测试数据, 输出一个浮点数, 表示该组测试数据的答案, 答案保留 4 位小数, 每组输出占一行。

样例输入:

2

1

2

样例输出:

1.0000

1.6667

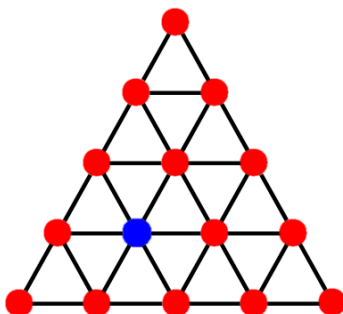
Problem H: 董与黑的碰撞-陷阵

时间限制: 1000MS

描述:

火霸的易容术和分身术接连被智慧的地球人破功, 董先森从分身中找到了火霸, 并劝其放弃抵抗。再次恼羞成怒的火霸使出了浑身解数, 利用分身形成了一个 n 阶正三角形的阵列, 并将董先森困在了其中某个位置, 董先森能看到的每一个分身都将与他进行对抗。

以下图为例, 这是一个 4 阶的正三角形阵列, 蓝点表示董先森, 红点表示火霸的分身, 阵列中共有 14 个分身, 而董先森能看到其中 12 个。



为了方便表示点的位置, 现在给每个点定义一个坐标 (x,y) , 其中 x 表示该点与三角阵最左面一条边的距离, y 表示该点与三角阵最下面一条边的距离。上图董先森的坐标即为 $(1,1)$, 董先森看不到的两个点的坐标分别为 $(3,1)$ 和 $(1,3)$ 。

现在告诉你 n 的大小, 以及董先森的坐标 (x,y) , 请你求出董先森所需要对抗的分身数量。

输入:

首先输入一个正整数 T , 表示测试数据的组数。对于每组数据, 输入三个正整数 n, x, y 。

数据范围:

$$1 \leq T \leq 1000$$

$$1 \leq n \leq 1000000$$

$$0 \leq x+y \leq n$$

输出:

对于每组测试数据, 输出一个整数, 表示该组测试数据的答案, 每组输出占一行。

样例输入:

3

1 0 0

2 0 0

4 1 1

样例输出：

2

3

12

Problem I: 董与黑的碰撞-决战

时间限制: 2000MS

描述:

被围困的董先森与火霸的众多分身展开了最终决战! 战斗一打响, 火霸就暴露了业余 ACMer 的本质, 因为火霸施展的每个技能, 在宇宙的数据库中都已记录, 只要通过火霸打出的一个片段, 就能检索出这个片段属于哪个技能, 从而各个击破。聪明的你赶快帮助董先森完成这最终一战吧!

为了方便输入, 我们给每个招式定义一个编号, 每个技能则包含一套有序的招式。一个片段属于某个技能, 当且仅当它是这个技能的一个连续的子序列, 例如:

技能 1: "baoli", 包含招式 1, 2, 3

技能 2: "kashi", 包含招式 3, 4, 5

技能 3: "suiji", 包含招式 5, 3, 2

如果火霸打出了片段 2, 3, 则可以得出火霸在施展"baoli"技能, 如果打出招式片段 4, 5, 则是"kashi"技能, 如果打出片段 3, 5, 则不属于任何技能。

现在告诉你数据库中共有 n 个技能, 第 i 个技能的名称为 $s[i]$, 包含了 $c[i]$ 个招式, 分别是 $a[i,1]$, $a[i,2]$, ..., $a[i,c[i]]$ 。

同时告诉你火霸打出的 m 个片段, 第 i 个片段包含了 $d[i]$ 个招式, 分别是 $b[i,1]$, $b[i,2]$, ..., $b[i,d[i]]$ 。请你求出火霸打出的每个片段属于哪个技能。

输入:

本题只包含一组测试数据。

对于每组数据, 第一行输入两个正整数 n, m , 表示有 n 个技能和 m 个片段。

接下来 n 行, 依次输入第 $1 \sim n$ 号技能, 每行依次输入技能的名称 $s[i]$ 、技能包含的招式数量 $c[i]$ 、技能包含的招式编号 $a[i,1]$, $a[i,2]$, ..., $a[i,c[i]]$ 。

接下来 m 行, 依次输入 m 个片段, 每行依次输入片段包含的招式数量 $d[i]$ 、片段包含的招式编号 $b[i,1]$, $b[i,2]$, ..., $b[i,d[i]]$ 。

数据范围:

$1 \leq n, m \leq 1000$

$1 \leq \text{strlen}(s[i]) \leq 10$, $s[i]$ 中只包含小写字母

$1 \leq c[i], d[i] \leq 1000$

$1 \leq a[i,j], b[i,j] \leq 20$

输出:

对于每个片段, 输出对应的技能名称。

如果某个片段对应多个技能, 输出技能编号最小的那一个。

如果某个片段没有对应的技能, 输出 "shenmegui" (不包含引号)。

每组输出占一行。

样例输入：

```
3 3
baoli 3 1 2 3
kashi 3 3 4 5
suiji 3 5 3 2
2 2 3
2 4 5
2 3 5
```

样例输出：

```
baoli
kasha
shenmegui
```


Problem J: 董与黑的碰撞-彩蛋

时间限制: 1000MS

描述:

公元 8102 年, 第 1542 届世界杯在火星举办, 结束战斗后的董先森来到火星练习射门。火星上的球门是三角形的, 安放在一张坐标纸上。董先森把球打到坐标纸上的一个点上。他希望你能帮他判断他打进了多少次 (打在门框上只能弹到门外)。

输入:

首先输入一个正整数 T , 表示测试数据的组数。

对于每组样例的前三行, 每行有两个整数 x, y , 表示球门上的一个顶点坐标为 (x, y) 。

第四行有一个整数 n , 表示董先森总共打了 n 脚。

接下来的 n 行, 每行有两个整数 x, y , 表示球的落点。

数据范围:

$1 \leq T \leq 1000$

$1 \leq n \leq 1000$

$-1000 \leq x \leq 1000$

$-1000 \leq y \leq 1000$

输出:

对于每组测试数据, 输出一个整数, 表示董先森打进的次数, 每组输出占一行。

样例输入:

```
0 4
4 0
0 0
4
0 0
1 1
2 2
3 3
```

样例输出:

```
1
```