第 31 届全国信息学奥林匹克竞赛

CCF NOI 2014

第一试

竞赛时间：2014 年 7 月 27 日 8:00-13:00

题目名称	起床困难综合症	魔法森林	消除游戏
目录	sleep	forest	game
可执行文件名	sleep	forest	game
输入文件名	sleep.in	forest.in	game1.in~game10.in
输出文件名	sleep.out	forest.out	game1.out~game10.out
每个测试点时限	1 秒	3 秒	N/A
内存限制	512MB	512MB	N/A
测试点数目	10	20	10
每个测试点分值	10	5	10
是否有部分分	否	否	是
题目类型	传统型	传统型	提交答案型
是否有附加文件	是	是	是

提交源程序须加后缀

对于 Pascal 语言	sleep.pas	forest.pas	N/A
对于 C　　语言	sleep.c	forest.c	N/A
对于 C++　语言	sleep.cpp	forest.cpp	N/A

注意：最终测试时，所有编译命令均不打开任何优化开关。

起床困难综合症

【问题描述】

21 世纪，许多人得了一种奇怪的病：起床困难综合症，其临床表现为：起床难，起床后精神不佳。作为一名青春阳光好少年，atm 一直坚持与起床困难综合症作斗争。通过研究相关文献，他找到了该病的发病原因：在深邃的太平洋海底中，出现了一条名为 drd 的巨龙，它掌握着睡眠之精髓，能随意延长大家的睡眠时间。正是由于 drd 的活动，起床困难综合症愈演愈烈，以惊人的速度在世界上传播。为了彻底消灭这种病，atm 决定前往海底，消灭这条恶龙。

历经千辛万苦，atm 终于来到了 drd 所在的地方，准备与其展开艰苦卓绝的战斗。drd 有着十分特殊的技能，他的防御战线能够使用一定的运算来改变他受到的伤害。具体说来，drd 的防御战线由 n 扇防御门组成。每扇防御门包括一个运算 op 和一个参数 t，其中运算一定是 `OR,XOR,AND` 中的一种，参数则一定为非负整数。如果还未通过防御门时攻击力为 x，则其通过这扇防御门后攻击力将变为 x op t。最终 drd 受到的伤害为对方初始攻击力 x **依次经过所有 n 扇防御门**后转变得到的攻击力。

由于 atm 水平有限，他的初始攻击力只能为 0 到 m 之间的一个整数（即他的初始攻击力只能在 $0, 1, \ldots, m$ 中任选，但在通过防御门之后的攻击力不受 m 的限制）。为了节省体力，他希望通过选择合适的初始攻击力使得他的攻击能让 drd 受到最大的伤害，请你帮他计算一下，他的一次攻击最多能使 drd 受到多少伤害。

【输入格式】

从文件 ***sleep.in*** 中读入数据。

输入文件的第 1 行包含 2 个整数，依次为 n, m，表示 drd 有 n 扇防御门，atm 的初始攻击力为 0 到 m 之间的整数。

接下来 n 行，依次表示每一扇防御门。每行包括一个字符串 op 和一个非负整数 t，两者由一个空格隔开，且 op 在前，t 在后，op 表示该防御门所对应的操作，t 表示对应的参数。

【输出格式】

输出到文件 ***sleep.out*** 中。

输出一行一个整数，表示 atm 的一次攻击最多使 drd 受到多少伤害。

【样例输入 1】

```
3 10
AND 5
OR 6
XOR 7
```

【样例输出 1】

```
1
```

【样例说明 1】

atm 可以选择的初始攻击力为 0,1, … ,10。

假设初始攻击力为 4，最终攻击力经过了如下计算

```
4 AND 5 = 4
4 OR 6 = 6
6 XOR 7 = 1
```

类似的，我们可以计算出初始攻击力为 1,3,5,7,9 时最终攻击力为 0，初始攻击力为 0,2,4,6,8,10 时最终攻击力为 1，因此 atm 的一次攻击最多使 drd 受到的伤害值为 1。

【样例输入输出 2】

见选手目录下的 ***sleep/sleep.in*** 与 ***sleep/sleep.ans***。

【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	n, m 的规模	约定	备注
1	$2 \le n \le 100, m = 0$	$0 \le t \le 10^9$ op 一定为 OR, XOR, AND 中的一种	
2	$2 \le n \le 1{,}000$ $1 \le m \le 1{,}000$		
3			
4	$2 \le n, m \le 10^5$		存在一扇防御门为 AND 0
5			所有防御门的操作均相同
6			
7	$2 \le n \le 10^5$ $2 \le m \le 10^9$		所有防御门的操作均相同
8			
9			
10			

【运算解释】

在本题中，<u>选手需要先将数字变换为二进制后再进行计算</u>。如果操作的两个数二进制长度不同，则在前补 0 至相同长度。

OR 为按位或运算，处理两个长度相同的二进制数，两个相应的二进制位中只要有一个为 1，则该位的结果值为 1，否则为 0。XOR 为按位异或运算，对等长二进制模式或二进制数的每一位执行逻辑异或操作。如果两个相应的二进制位不同（相异），则该位的结果值为 1，否则该位为 0。 AND 为按位与运算，处理两个长度相同的二进制数，两个相应的二进制位都为 1，该位的结果值才为 1，否则为 0。

例如，我们将十进制数 5 与十进制数 3 分别进行 OR，XOR 与 AND 运算，可以得到如下结果：

```
  0101 (十进制 5)      0101 (十进制 5)      0101 (十进制 5)
OR 0011 (十进制 3)   XOR 0011 (十进制 3)   AND 0011 (十进制 3)
 = 0111 (十进制 7)     = 0110 (十进制 6)     = 0001 (十进制 1)
```

魔法森林

【问题描述】

为了得到书法大家的真传，小 E 同学下定决心去拜访住在魔法森林中的隐士。魔法森林可以被看成一个包含 n 个节点 m 条边的无向图，节点标号为 $1,2,3,\ldots,n$，边标号为 $1,2,3,\ldots,m$。初始时小 E 同学在 1 号节点，隐士则住在 n 号节点。小 E 需要通过这一片魔法森林，才能够拜访到隐士。

魔法森林中居住了一些妖怪。每当有人经过一条边的时候，这条边上的妖怪就会对其发起攻击。幸运的是，在 1 号节点住着两种守护精灵：A 型守护精灵与 B 型守护精灵。小 E 可以借助它们的力量，达到自己的目的。

只要小 E 带上足够多的守护精灵，妖怪们就不会发起攻击了。具体来说，无向图中的每一条边 e_i 包含两个权值 a_i 与 b_i 。若身上携带的 A 型守护精灵个数**不少于 a_i** ，且 B 型守护精灵个数**不少于 b_i** ，这条边上的妖怪就不会对通过这条边的人发起攻击。**当且仅当通过这片魔法森林的过程中没有任意一条边的妖怪向小 E 发起攻击，他才能成功找到隐士。**

由于携带守护精灵是一件非常麻烦的事，小 E 想要知道，要能够成功拜访到隐士，最少需要携带守护精灵的总个数。**守护精灵的总个数**为 A 型守护精灵的个数与 B 型守护精灵的个数之和。

【输入格式】

从文件 ***forest.in*** 中读入数据。

输入文件的第 1 行包含两个整数 n, m，表示无向图共有 n 个节点，m 条边。

接下来 m 行，第 $i+1$ 行包含 4 个正整数 X_i, Y_i, a_i, b_i，描述第 i 条无向边。其中 X_i 与 Y_i 为该边两个端点的标号， a_i 与 b_i 的含义如题所述。

注意数据中可能包含重边与自环。

【输出格式】

输出到文件 ***forest.out*** 中。

输出一行一个整数：如果小 E 可以成功拜访到隐士，输出小 E 最少需要携带的**守护精灵的总个数**；如果无论如何小 E 都无法拜访到隐士，输出“-1”（不含引号）。

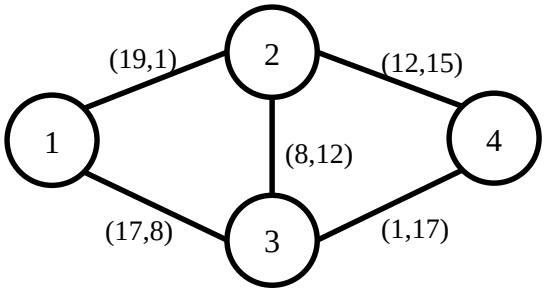
【样例输入 1】

```
4 5
1 2 19 1
2 3 8 12
2 4 12 15
1 3 17 8
3 4 1 17
```

【样例输出 1】

```
32
```

【样例说明 1】



如果小 E 走路径 1→2→4，需要携带 19+15=34 个守护精灵；

如果小 E 走路径 1→3→4，需要携带 17+17=34 个守护精灵；

如果小 E 走路径 1→2→3→4，需要携带 19+17=36 个守护精灵；

如果小 E 走路径 1→3→2→4，需要携带 17+15=32 个守护精灵。

综上所述，小 E 最少需要携带 32 个守护精灵。

【样例输入 2】

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

【样例输出 2】

```
-1
```

【样例说明 2】

小 E 无法从 1 号节点到达 3 号节点，故输出-1。

【样例输入输出 3】

见选手目录下的 ***forest/forest.in*** 与 ***forest/forest.ans***。

【数据规模与约定】

<table>
<tr><th>测试点编号</th><th>n</th><th>m</th><th>a_i, b_i</th></tr>
<tr><td>1</td><td rowspan="3">$2 \leq n \leq 5$</td><td rowspan="3">$0 \leq m \leq 10$</td><td rowspan="3">$1 \leq a_i, b_i \leq 10$</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td rowspan="3">$2 \leq n \leq 500$</td><td rowspan="3">$0 \leq m \leq 3{,}000$</td><td rowspan="6">$1 \leq a_i, b_i \leq 50{,}000$</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td><td rowspan="4">$2 \leq n \leq 5{,}000$</td><td rowspan="4">$0 \leq m \leq 10{,}000$</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td><td rowspan="10">$2 \leq n \leq 50{,}000$</td><td rowspan="10">$0 \leq m \leq 100{,}000$</td><td rowspan="4">$1 \leq a_i \leq 30$
$1 \leq b_i \leq 50{,}000$</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td rowspan="6">$1 \leq a_i, b_i \leq 50{,}000$</td></tr>
<tr><td>16</td></tr>
<tr><td>17</td></tr>
<tr><td>18</td></tr>
<tr><td>19</td></tr>
<tr><td>20</td></tr>
</table>

消除游戏

【问题描述】

最近，小 Z 迷上了一款新型消除游戏。这款游戏在一个 $n \times m$ 的方格中进行。初始时方格中均为 $0 \sim 9$ 的整数。进行消除后方格中会出现空白，用 -1 表示。为了方便，我们将第 i 行，第 j 列的数记为 $A_{i,j}$，并将其坐标记为 (i,j)。

给定三个参数 l_{min}, l_{max} 以及 K，玩家可以进行**不超过** K 次操作。对于每次操作，玩家需要在方格中找到一条长度为 l 的路径。形式化地，该路径用两个长度为 l 的序列 $x_1, x_2, \ldots, x_l$ 和 $y_1, y_2, \ldots, y_l$ 表示，需要满足如下条件：

1. $1 \le x_i \le n, 1 \le y_i \le m$，其中$1 \le i \le l$，即 (x_i, y_i) 对应于方格中的一个合法位置；
2. $|x_i - x_{i+1}| + |y_i - y_{i+1}| = 1$，其中$1 \le i < l$，即 (x_i, y_i) 与 (x_{i+1}, y_{i+1}) 是方格中相邻的两个位置；
3. $x_i \ne x_j$ 或 $y_i \ne y_j$，其中$1 \le i < j \le l$，即路径不能经过重复的格子；
4. $A_{x_i,y_i} \ne -1$，其中$1 \le i \le l$，即路径不能经过空白的格子；
5. $A_{x_1,y_1} \ne 0$，即路径不能以数字 0 为起点；
6. $l_{min} \le l \le l_{max}$，即路径的长度需要在给定的范围内。

将路径上的数字串成一个整数 N，形式化地，

$$N = \sum_{i=1}^{l} A_{x_i,y_i} \times 10^{l-i}$$

游戏会给出两个参数 c_1, c_2 用于计算玩家本次操作的得分：

1. 如果数 N 是质数，那么将获得**质数得分** l^{c_1}，否则获得**质数得分** 1。
2. 如果数 N 是回文数（即，将数 N 的十进制表达看成一个字符串，这个字符串的逆序串和它本身完全相同），那么将获得**回文数得分** l^{c_2}，否则获得**回文数得分** 1。
3. 如果**质数得分**和**回文数得分**均为 1，那么**本次操作的得分**为 0；否则**本次操作的得分**为**质数得分与回文数得分**之和。

每次操作过后，若**该次操作的得分**等于 0，那么你浪费了一次操作机会，而局面不会有任何改变。若**该次操作的得分**大于 0，则将路径上的数替换为空白，并使空白上方的数字垂直下落。形式化地，执行以下操作：

1. 执行 $A_{x_i,y_i} \leftarrow -1$，其中$1 \le i \le l$。
2. 枚举所有格子。如果存在某个格子(i,j)，满足 $i \ne n$，$A_{i,j} \ne -1, A_{i+1,j} = -1$，执行 $A_{i+1,j} \leftarrow A_{i,j}$，$A_{i,j} \leftarrow -1$。反复执行这个操作直到方格中不再存在这样的格子。

下面举例说明玩家操作和数字消除的具体情况。某次游戏中，$n = m = 3$，$c_1 = c_2 = 1$，玩家面临如图 1 的局面：

2	-1	-1
2	3	3
4	**7**	1

图 1

2	-1	-1
2	3	3
-1	**-1**	1

图 2

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

图 3

1. 玩家选择了包含格子 $4,7$ 的路径。计算可得 $N=47$。由于 47 是质数，其**质数得分**为 $l^{c_1}=2^1=2$；由于 47 不是回文数，其**回文数得分**为 1。于是，**该次操作得分**为 $2+1=3$。
2. 由于本次操作得分非零，路径上的数字将变为空白（如图 2）。
3. 空白上方的数字垂直下落（如图 3），此时玩家方可进行下一次操作。

我们还会给你一个参数 F，在所有操作完成后，玩家的**最终得分**的计算方式由 F 决定：如果 F 取值为 0，那么玩家的最终得分为所有操作的分数总和；如果 F 取值为 1，那么玩家的最终得分为所有操作的分数总和除以 $\mathbf{2^d}$ 后向下取整，即

$$\text{最终得分}=\begin{cases}\text{所有操作的分数总和}, & F=0\\ \left\lfloor\dfrac{\text{所有操作的分数总和}}{2^d}\right\rfloor, & F=1\end{cases}$$

其中 d 为最终方格中非空白格子的数目。

小 Z 沉迷于这个有趣的游戏中不能自拔。她想请你帮助，针对给定的输入参数，给出游戏局面的操作方案。当然，**最终得分**越大越好。

【输入格式】

所有输入数据 ***game1.in~game10.in*** 已在试题目录下。

输入的第 1 行包含 8 个用空格分隔的整数 $n,m,K,l_{min},l_{max},c_1,c_2,F$，含义同题面描述。

随后 n 行，每行 m 个整数，表示方格 A。数之间用一个空格分隔。

输入文件中不会包含多余的空行，行末不会存在多余的空格。

【输出格式】

针对给定的 10 个输入文件 ***game1.in~game10.in***，你需要分别提交你的输出文件 ***game1.out~game10.out***。

输出文件第 1 行为一个整数 $M(0\le M\le K)$，为你的操作次数。

随后，输出文件还应包含 M 行，每行描述一次操作。对于每一行，最开始的整数 l 表示这次操作中选定路径的长度。接下来有 $2l$ 个数字，分别为 $x_1,y_1,x_2,y_2,\ldots,x_l,y_l$。

输出文件中不应包含多余的空格和空行。一行的多个整数之间使用一个空格分隔。

输出文件大小不能超过 1 MB。数据保证一个合法的输出文件大小不会超过这个上界。

【样例输入 1】

```
3 3 100 2 3 1 1 0
2 1 1
2 3 3
4 7 1
```

【样例输出 1】

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

【样例说明 1】

4次消除得到的数与相应的分数分别是：37，得分为 $2+1=3$；41，得分为 $2+1=3$；22，得分为 $1+2=3$；131，得分为 $3+3=6$。总共得分为 15。可能存在更优的方案。

【样例输入 2】

```
1 3 100 2 3 1 1 1
2 1 1
```

【样例输出 2】

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

【样例说明 2】

本方案仅一次消除操作。消除的数为 11，本次操作得分为 $2+2=4$ 。由于 $F=1$，最终得分为每次操作得分之和 4 除以 $2^1=2$ 后下取整，为 2 。若选择消除路径 211 ，则会得到本局面最佳分数 4 。

【评分方式】

对于每组数据，我们设置了 9 个评分参数 $a_{10}, a_9, a_8, \ldots, a_2$。如果选手的输出不合法，则得零分。否则，在你的方案中，若游戏得分为 w_{user} ，你的分数将会由下表给出：

得分	条件	得分	条件
10	$w_{user} \geq a_{10}$	5	$w_{user} \geq a_5$
9	$w_{user} \geq a_9$	4	$w_{user} \geq a_4$
8	$w_{user} \geq a_8$	3	$w_{user} \geq a_3$
7	$w_{user} \geq a_7$	2	$w_{user} \geq a_2$
6	$w_{user} \geq a_6$	1	$w_{user} > 0$

【如何测试你的输出】

在终端中先切换到该试题的目录下

```
cd game
```

我们提供 ***checker*** 这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是，在终端中运行

```
./checker <case_no>
```

其中 `case_no` 是测试数据的编号。例如

```
./checker 3
```

将测试 `game3.out` 是否可以接受。

在你调用这个程序后，***checker*** 将根据你给出的输出文件给出测试的结果，其中包括：

1. 非法退出：未知错误
2. Output file do not exist.：找不到输出文件
3. Output invalid!：输出文件有误，此时可能包含具体错误信息
4. Details: xxx.：其他提示信息
5. Correct! Your score is x.：输出可接受，你的得分为x

【提示】

请妥善保存输入文件 ***game*.in*** 和输出文件 ***game*.out***，及时备份，以免误删。