第 34 届全国青少年信息学奥林匹克竞赛

CCF NOI 2017

第二试

时间：2017 年 7 月 21 日 08:00 ~ 13:00

题目名称	游戏	蔬菜	分身术
题目类型	传统型	传统型	传统型
目录	game	vegetables	phantom
可执行文件名	game	vegetables	phantom
输入文件名	game.in	vegetables.in	phantom.in
输出文件名	game.out	vegetables.out	phantom.out
每个测试点时限	1.0 秒	3.0 秒	3.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	25	20
每个测试点分值	5	4	5

提交源程序文件名

对于 C++ 语言	game.cpp	vegetables.cpp	phantom.cpp
对于 C 语言	game.c	vegetables.c	phantom.c
对于 Pascal 语言	game.pas	vegetables.pas	phantom.pas

编译选项

对于 C++ 语言	-lm	-O2 -lm	-O2 -lm
对于 C 语言	-lm	-O2 -lm	-O2 -lm
对于 Pascal 语言		-O2	-O2

游戏（game）

【题目背景】

狂野飙车是小 L 最喜欢的游戏。与其他业余玩家不同的是，小 L 在玩游戏之余，还精于研究游戏的设计，因此他有着与众不同的游戏策略。

【题目描述】

小 L 计划进行 n 场游戏，每场游戏使用一张地图，小 L 会选择一辆车在该地图上完成游戏。

小 L 的赛车有三辆，分别用大写字母 A、B、C 表示。地图一共有四种，分别用小写字母 x、a、b、c 表示。其中，赛车 A 不适合在地图 a 上使用，赛车 B 不适合在地图 b 上使用，赛车 C 不适合在地图 c 上使用，而地图 x 则适合所有赛车参加。适合所有赛车参加的地图并不多见，最多只会有 d 张。

n 场游戏的地图可以用一个小写字母组成的字符串描述。例如：S=xaabxcbc 表示小 L 计划进行 8 场游戏，其中第 1 场和第 5 场的地图类型是 x，适合所有赛车，第 2 场和第 3 场的地图是 a，不适合赛车 A，第 4 场和第 7 场的地图是 b，不适合赛车 B，第 6 场和第 8 场的地图是 c，不适合赛车 C。

小 L 对游戏有一些特殊的要求，这些要求可以用四元组 (i, h_i, j, h_j) 来描述，表示若在第 i 场使用型号为 h_i 的车子，则第 j 场游戏要使用型号为 h_j 的车子。

你能帮小 L 选择每场游戏使用的赛车吗？如果有多种方案，输出任意一种方案。如果无解，输出 “-1”（不含双引号）。

【输入格式】

从文件 ***game.in*** 中读入数据。

输入第一行包含两个非负整数 n, d。

输入第二行为一个字符串 S 。

n, d, S 的含义见题目描述，其中 S 包含 n 个字符，且其中恰好 d 个为小写字母 x。

输入第三行为一个正整数 m ，表示有 m 条用车规则。接下来 m 行，每行包含一个四元组 i, h_i, j, h_j ，其中 i, j 为整数，h_i, h_j 为字符 a 、b 或 c，含义见题目描述。

【输出格式】

输出到文件 ***game.out*** 中。

输出一行。

若无解输出 “-1”（不含双引号）。

若有解，则包含一个长度为 n 的仅包含大写字母 A、B、C 的字符串，表示小 L 在这 n 场游戏中如何安排赛车的使用。如果存在多组解，输出其中任意一组即可。

【样例 1 输入】

```
3 1
xcc
1
1 A 2 B
```

【样例 1 输出】

```
ABA
```

【样例 2】

见选手目录下的 *game/game2.in* 与 *game/game2.ans*。

【样例 1 解释】

小 L 计划进行 3 场游戏，其中第 1 场的地图类型是 x，适合所有赛车，第 2 场和第 3 场的地图是 c，不适合赛车 C。

小 L 希望：若第 1 场游戏使用赛车 A，则第 2 场游戏使用赛车 B。

那么为这 3 场游戏分别安排赛车 A、B、A 可以满足所有条件。

若依次为 3 场游戏安排赛车为 BBB 或 BAA 时，也可以满足所有条件，也被视为正确答案。但依次安排赛车为 AAB 或 ABC 时，因为不能满足所有条件，所以不被视为正确答案。

【子任务】

<table>
<tr><th>测试点编号</th><th>n</th><th>d</th><th>m</th><th>其他性质</th></tr>
<tr><td>1</td><td rowspan="2">≤ 2</td><td>0</td><td rowspan="2">≤ 4</td><td rowspan="6">无</td></tr>
<tr><td>2</td><td>$\le n$</td></tr>
<tr><td>3</td><td rowspan="2">≤ 5</td><td>0</td><td rowspan="2">≤ 10</td></tr>
<tr><td>4</td><td>$\le n$</td></tr>
<tr><td>5</td><td rowspan="2">≤ 10</td><td>0</td><td rowspan="2">≤ 20</td></tr>
<tr><td>6</td><td>≤ 8</td></tr>
<tr><td>7</td><td rowspan="4">≤ 20</td><td rowspan="2">0</td><td rowspan="4">≤ 40</td><td>S 中只包含 c</td></tr>
<tr><td>8</td><td>无</td></tr>
<tr><td>9</td><td rowspan="2">≤ 8</td><td>S 中只包含 x 或 c</td></tr>
<tr><td>10</td><td>无</td></tr>
<tr><td>11</td><td rowspan="4">≤ 100</td><td rowspan="2">0</td><td rowspan="4">≤ 200</td><td>S 中只包含 c</td></tr>
<tr><td>12</td><td>无</td></tr>
<tr><td>13</td><td rowspan="2">≤ 8</td><td>S 中只包含 x 或 c</td></tr>
<tr><td>14</td><td rowspan="2">无</td></tr>
<tr><td>15</td><td rowspan="3">≤ 5000</td><td>0</td><td rowspan="3">≤ 10000</td></tr>
<tr><td>16</td><td rowspan="2">≤ 8</td><td>S 中只包含 x 或 c</td></tr>
<tr><td>17</td><td rowspan="2">无</td></tr>
<tr><td>18</td><td rowspan="3">≤ 50000</td><td>0</td><td rowspan="3">≤ 100000</td></tr>
<tr><td>19</td><td rowspan="2">≤ 8</td><td>S 中只包含 x 或 c</td></tr>
<tr><td>20</td><td>无</td></tr>
</table>

蔬菜（vegetables）

【题目描述】

小 N 是蔬菜仓库的管理员，负责设计蔬菜的销售方案。

在蔬菜仓库中，共存放有 n 种蔬菜，小 N 需要根据不同蔬菜的特性，综合考虑各方面因素，设计合理的销售方案，以获得最多的收益。

在计算销售蔬菜的收益时，每销售一个单位第 i 种蔬菜，就可以获得 a_i 的收益。

特别地，由于政策鼓励商家进行多样化销售，第一次销售第 i 种蔬菜时，还会额外得到 s_i 的额外收益。

在经营开始时，第 i 种蔬菜的库存为 c_i 个单位。

然而，蔬菜的保鲜时间非常有限，一旦变质就不能进行销售，不过聪明的小 N 已经计算出了每个单位蔬菜变质的时间：对于第 i 种蔬菜，存在保鲜值 x_i，每天结束时会有 x_i 个单位的蔬菜变质，直到所有蔬菜都变质。（注意：每一单位蔬菜的变质时间是固定的，不随销售发生变化）

形式化地：对于所有的满足条件 $d \times x_i \le c_i$ 的正整数 d，有 x_i 个单位的蔬菜将在第 d 天结束时变质。

特别地，若 $(d-1) \times x_i \le c_i < d \times x_i$，则有 $c_i - (d-1) \times x_i$ 单位的蔬菜将在第 d 天结束时变质。

注意，当 $x_i = 0$ 时，意味着这种蔬菜不会变质。

同时，每天销售的蔬菜**总量**也是有限的，最多不能超过 m 个单位。

现在，小 N 有 k 个问题，想请你帮忙算一算。每个问题的形式都是：对于已知的 p_j，如果需要销售 p_j 天，最多能获得多少收益？

【输入格式】

从文件 ***vegetables.in*** 中读入数据。

第一行包含三个正整数 n, m, k，分别表示蔬菜的种类数目、每天能售出蔬菜总量上限、小 N 提出的问题的个数。

接下来 n 行，每行输入四个非负整数，描述一种蔬菜的特点，依次为 a_i, s_i, c_i, x_i，意义如上文所述。

接下来 k 行，每行输入一个非负整数 p_j，意义如上文所述。

【输出格式】

输出到文件 ***vegetables.out*** 中。

输出 k 行，每行包含一个整数，第 i 行的数表示第 i 个问题的答案。

【样例 1 输入】

```
2 3 2
3 3 3 3
2 5 8 3
1
3
```

【样例 1 输出】

```
16
27
```

【样例 2】

见选手目录下的 ***vegetables/vegetables2.in*** 与 ***vegetables/vegetables2.ans***。

【样例 3】

见选手目录下的 ***vegetables/vegetables3.in*** 与 ***vegetables/vegetables3.ans***。

【样例 1 解释】

共有两种蔬菜：

销售第 1 种蔬菜时，每销售一单位可以获得的收益为 3 ，第一次销售这种蔬菜时，额外可以获得的收益为 3。这种蔬菜共有 3 个单位，均会在第一天结束时变质。

销售第 2 种蔬菜时，每销售一单位可以获得的收益为 2 ，第一次销售这种蔬菜时，额外可以获得的收益为 5。这种蔬菜共有 8 个单位，其中，有 3 单位在第一天结束时变质，3 单位在第二天结束时变质，2 单位在第三天结束时变质。

在只销售 1 天时，应当销售 2 单位的第一种蔬菜和 1 单位的第二种蔬菜。

在这种情况下：销售第一种蔬菜的收益为 $2\times3+3$ ；销售第二种蔬菜的收益为 $1\times2+5$ ；总共获得的收益为 $(2\times3+3)+(1\times2+5)=16$ 。

在只销售 3 天时，第一天应当销售 3 单位的第一种蔬菜，第二天应当销售 3 单位的第二种蔬菜（此时选择在第二天结束时会变质的 3 个单位出售），第三天销售 2 单位的第二种蔬菜。

在这种情况下：销售第一种蔬菜的收益为 $3\times3+3$ ；销售第二种蔬菜的收益为 $(3+2)\times2+5$ ；总共获得的收益为 $(3\times3+3)+[(3+2)\times2+5]=27$ 。

【子任务】

<table>
<tr><th>测试点编号</th><th>n</th><th>m</th><th>p_j</th><th>特性 1</th><th>特性 2</th></tr>
<tr><td>1</td><td>≤ 2</td><td rowspan="6">≤ 10</td><td rowspan="3">$\le 10^3$</td><td rowspan="11">无</td><td rowspan="11">无</td></tr>
<tr><td>2</td><td>≤ 3</td></tr>
<tr><td>3</td><td>≤ 4</td></tr>
<tr><td>4</td><td rowspan="3">$\le 10^3$</td><td>≤ 2</td></tr>
<tr><td>5</td><td>≤ 3</td></tr>
<tr><td>6</td><td rowspan="2">≤ 4</td></tr>
<tr><td>7</td><td>≤ 4</td><td>≤ 1</td></tr>
<tr><td>8</td><td>≤ 6</td><td>≤ 2</td><td>≤ 6</td></tr>
<tr><td>9</td><td>≤ 8</td><td>≤ 1</td><td>≤ 8</td></tr>
<tr><td>10</td><td>≤ 10</td><td>≤ 2</td><td>≤ 10</td></tr>
<tr><td>11</td><td>≤ 20</td><td>≤ 3</td><td>≤ 20</td></tr>
<tr><td>12</td><td rowspan="4">$\le 10^2$</td><td rowspan="14">≤ 10</td><td rowspan="4">$\le 10^2$</td><td>有</td><td>无</td></tr>
<tr><td>13</td><td rowspan="3">无</td><td>有</td></tr>
<tr><td>14</td><td rowspan="2">无</td></tr>
<tr><td>15</td></tr>
<tr><td>16</td><td rowspan="5">$\le 10^3$</td><td rowspan="5">$\le 10^3$</td><td rowspan="2">有</td><td>有</td></tr>
<tr><td>17</td><td>无</td></tr>
<tr><td>18</td><td rowspan="3">无</td><td>有</td></tr>
<tr><td>19</td><td rowspan="2">无</td></tr>
<tr><td>20</td></tr>
<tr><td>21</td><td rowspan="5">$\le 10^5$</td><td rowspan="5">$\le 10^5$</td><td rowspan="2">有</td><td>有</td></tr>
<tr><td>22</td><td>无</td></tr>
<tr><td>23</td><td rowspan="3">无</td><td>有</td></tr>
<tr><td>24</td><td rowspan="2">无</td></tr>
<tr><td>25</td></tr>
</table>

特性 1：所有的 s_i 均为 0。

特性 2：所有的 x_i 均为 0。

对于所有的测试数据，均保证 k 组询问中的 p_j 互不相同。

对于所有的测试数据，均保证 $0 < a_i, c_i \le 10^9$，$0 \le s_i, x_i \le 10^9$。

分身术（phantom）

【题目描述】

“分！身！术！”—— 小 P

平面上有 n 个小 P 的分身。定义一组分身占领的区域为覆盖这组分身的最小凸多边形。小 P 能力有限，每一时刻都会有若干分身消失。但在下一时刻之前，小 P 会使用

“分！身！术！”

使得这些消失的分身重新出现在原来的位置。小 P 想知道，每一时刻分身消失后，剩下的分身占领的区域面积是多少？

【输入格式】

从文件 ***phantom.in*** 中读入数据。

输入第一行包含两个正整数 n, m，描述初始时分身的个数，和总时刻数。

接下来 n 行，第 i 行有两个整数 x_i, y_i ，描述第 i 个分身的位置。

接下来 m 行，每行的第一个整数 k 表示这一时刻有 k 个分身消失。接下来有 k 个非负整数 $c_1, c_2, \cdots c_k$，用于生成消失的分身的编号。

生成方式如下：

设上一个时刻中，分身占领面积的**两倍**为 S。则该时刻消失的分身 $p_1, p_2, \ldots, p_k$ 的编号为：

$$p_i = [(S + c_i) \bmod n] + 1$$

特别的，在第一个时刻，我们认为上一个时刻中，$S = -1$ ，即：第一个时刻消失的分身 $p_1, p_2, \ldots, p_k$ 的编号为：

$$p_i = [(-1 + c_i) \bmod n] + 1$$

【输出格式】

输出到文件 ***phantom.out*** 中。

按给出时刻的顺序依次输出 m 行，每行一个整数，表示该时刻剩余分身所占领区域面积的**两倍**。

【样例 1 输入】

```
6 2
-1 0
-1 -1
```

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

【样例 1 输出】

```
3
2
```

【样例 2】

见选手目录下的 ***phantom/phantom2.in*** 与 ***phantom/phantom2.ans***。

【样例 3】

见选手目录下的 ***phantom/phantom3.in*** 与 ***phantom/phantom3.ans***。

【样例 4】

见选手目录下的 ***phantom/phantom4.in*** 与 ***phantom/phantom4.ans***。

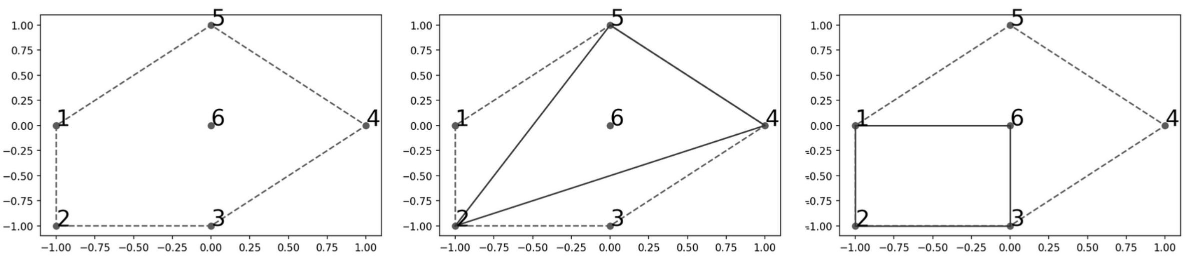
【样例 1 解释】

如下图所示：左图表示输入的 6 个分身的位置及它们占领的区域；中图表示第一个时刻的情形，消失的分身编号分别为 1,3,6，剩余 3 个点占领图中实线内部区域，占据面积的两倍为 3；右图表示第二个时刻的情形，消失的分身编号分别为

$$[(0+3) \bmod 6]+1=4$$

$$[(1+3) \bmod 6]+1=5$$

剩余的 4 个点占领图中实线内部区域。



【子任务】

<table>
<tr><th>测试点编号</th><th>$n \le$</th><th>$m \le$</th><th>k</th></tr>
<tr><td>1</td><td>10</td><td>10</td><td rowspan="4">$\le n-3$</td></tr>
<tr><td>2</td><td rowspan="3">1000</td><td rowspan="3">1000</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td rowspan="16">100000</td><td rowspan="16">100000</td><td rowspan="4">$=1$</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td rowspan="2">$=2$</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td><td>≤ 3</td></tr>
<tr><td>12</td><td>≤ 5</td></tr>
<tr><td>13</td><td>≤ 9</td></tr>
<tr><td>14</td><td>≤ 12</td></tr>
<tr><td>15</td><td>≤ 20</td></tr>
<tr><td>16</td><td rowspan="5">≤ 100</td></tr>
<tr><td>17</td></tr>
<tr><td>18</td></tr>
<tr><td>19</td></tr>
<tr><td>20</td></tr>
</table>

对于所有数据，保证：

- $|x_i|, |y_i| \le 10^8$；
- 没有两个分身的坐标是完全相同的；
- $k \le 100$；
- 所有时刻的 k 之和不超过 2×10^6；
- $0 \le c_i \le 2^{31} - 1$；
- 初始时，所有的 n 个分身占据区域面积大于 0；
- 定义所有 n 个分身所占据区域的**顶点集合**为 S，$|S| \ge 3$。在任意时刻，S 中至少存在两个未消失的分身。