试题一：杀蚂蚁

程序名：antbuster.pas/c/cpp
可执行文件名：antbuster.exe
输入文件名：antbuster.in
输出文件名：antbuster.out
时间限制：3s

最近，佳佳迷上了一款好玩的小游戏：antbuster。

游戏规则非常简单：在一张地图上，左上角是蚂蚁窝，右下角是蛋糕，蚂蚁会源源不断地从窝里爬出来，试图把蛋糕搬回蚂蚁窝。而你的任务，就是用原始资金以及杀蚂蚁获得的奖金造防御塔，杀掉这些试图跟你抢蛋糕的蚂蚁~

为了拿到尽可能高的分数，佳佳设计了很多种造塔的方案，但在尝试了其中的一小部分后，佳佳发现，这个游戏实在是太费时间了。为了节省时间，佳佳决定写个程序，对于每一种方案，模拟游戏进程，根据效果来判断方案的优劣。

根据自己在游戏中积累的一些经验，以及上网搜到的一些参数，佳佳猜了蚂蚁爬行的算法，并且假设游戏中的蚂蚁也是按这个规则选择路线：

1、每一秒钟开始的时候，蚂蚁都在平面中的某个整点上。如果蚂蚁没有扛着蛋糕，它会在该点留下 2 单位的信息素，否则它会留下 5 单位的信息素。然后蚂蚁会在正北、正南、正东、正西四个方向中选择一个爬过去。

2、选择方向的规则是：首先，爬完一个单位长度后到达的那个点上，不能有其他蚂蚁或是防御塔，并且那个点不能是蚂蚁上一秒所在的点（除非上一个时刻蚂蚁就被卡住，且这个时刻它仍无法动），当然，蚂蚁也不会爬出地图的边界（我们定义这些点为不可达点）。如果此时有多个选择，蚂蚁会选择信息素最多的那个点爬过去。

3、如果此时仍有多种选择，蚂蚁先面向正东，如果正东不是可选择的某个方向，它会顺时针转 90°，再次判断，如果还不是，再转 90°...直到找到可以去的方向。

4、如果将每只蚂蚁在洞口出现的时间作为它的活动时间的第 1 秒，那么每当这只蚂蚁的活动时间秒数为 5 的倍数的时候，它先按规则 1~3 确定一个方向，面对该方向后逆时针转 90°，若它沿当前方向会走到一个不可达点，它会不停地每次逆时针转 90°，直到它面对着一个可达的点，这样定下的方向才是蚂蚁最终要爬去的方向。

5、如果蚂蚁的四周都是不可达点，那么蚂蚁在这一秒内会选择停留在当前点。下一秒判断移动方向时，它上一秒所在点为其当前停留的点。

6、你可以认为蚂蚁在选定方向后，瞬间移动到它的目标点，这一秒钟剩下的时间里，它就停留在目标点。

7、蚂蚁按出生的顺序移动，出生得比较早的蚂蚁先移动。

然后，是一些有关地图的信息：

1、 每一秒，地图所有点上的信息素会损失 1 单位，如果那个点上有信息素的话。

2、 地图上某些地方是炮台。炮台的坐标在输入中给出。

3、 地图的长、宽在输入中给出，对于 n * m 的地图，它的左上角坐标为（0，0），右下角坐标为（n，m）。蚂蚁洞的位置为（0，0），蛋糕的位置为（n，m）。

4、 你可以把蚂蚁看做一个直径为 1 单位的圆，圆心位于蚂蚁所在的整点。

5、 游戏开始时，地图上没有蚂蚁，每个点上的信息素含量均为 0。

一些有关炮塔的信息：

1、 炮塔被放置在地图上的整点处。

2、 为了简单一些，我们认为这些炮塔都是激光塔。激光塔的射速是 1 秒/次，它的攻击伤害为 d/次，攻击范围为 r。你可以认为每秒蚂蚁移动完毕后，塔才开始攻击。并且，只有当代表蚂蚁的圆的圆心与塔的直线距离不超过 r 时，塔才算打得到那只蚂蚁。

3、 如果一只蚂蚁扛着蛋糕，那么它会成为 target，也就是说，任何打得到它的塔的炮口都会对准它。如果蛋糕好好地呆在原位，那么每个塔都会挑离它最近的蚂蚁进行攻击，如果有多只蚂蚁，它会选出生较早的一只。

4、 激光塔有个比较奇怪的特性：它在选定了打击目标后，只要目标在其射程内，塔到目标蚂蚁圆心的连线上的所有蚂蚁（这里“被打到”的判定变成了表示激光的线段与表示蚂蚁的圆有公共点）都会被打到并损 d 格血，但激光不会穿透它的打击目标打到后面的蚂蚁。

5、 尽管在真实游戏中，塔是可以升级的，但在这里我们认为塔的布局和等级就此定了下来，不再变动。

再介绍一下蚂蚁窝：

1、 如果地图上的蚂蚁不足 6 只，并且洞口没有蚂蚁，那么窝中每秒会爬出一只蚂蚁，直到地图上的蚂蚁数为 6 只。

2、 刚出生的蚂蚁站在洞口。

3、 每只蚂蚁有一个级别，级别决定了蚂蚁的血量，级别为 k 的蚂蚁的血量为 int(4*1.1^k)（int(x)表示对 x 取下整）。每被塔打一次，蚂蚁的血减少 d。注意，血量为 0 的蚂蚁仍能精力充沛地四处乱爬，只有一只蚂蚁的血被打成负数时，它才算挂了。

4、 蚂蚁的级别是这样算的：前 6 只出生的蚂蚁是 1 级，第 7~12 只是 2 级，依此类推。

最后给出关于蛋糕的介绍：

1、 简单起见，你可以认为此时只剩最后一块蛋糕了。如果有蚂蚁走到蛋糕的位置，并且此时蛋糕没有被扛走，那么这只蚂蚁就扛上了蛋糕。蚂蚁被打死后蛋糕归位。

2、 如果一只扛着蛋糕的蚂蚁走到蚂蚁窝的位置，我们就认为蚂蚁成功抢到了蛋糕，游戏结束。

3、 蚂蚁扛上蛋糕时，血量会增加 int（该蚂蚁出生时血量 / 2），但不会超过上限。

整理一下 1 秒钟内发生的事件：

1 秒的最初，如果地图上蚂蚁数不足 6，一只蚂蚁就会在洞口出生。

接着，蚂蚁们在自己所在点留下一些信息素后，考虑移动。先出生的蚂蚁先移动。

移动完毕后，如果有蚂蚁在蛋糕的位置上并且蛋糕没被拿走，它把蛋糕扛上，血量增加，并在这时被所有塔设成 target。

然后所有塔同时开始攻击。如果攻击结束后那只扛着蛋糕的蚂蚁挂了，蛋糕瞬间归位。

攻击结束后，如果发现扛蛋糕的蚂蚁没死并在窝的位置，就认为蚂蚁抢到了蛋糕。游戏也在此时结束。

最后，地图上所有点的信息素损失 1 单位。所有蚂蚁的年龄加 1。

漫长的 1 秒到此结束。

输入：

输入的第一行是 2 个用空格隔开的整数，n、m，分别表示了地图的长和宽。

第二行是 3 个用空格隔开的整数，s、d、r，依次表示炮塔的个数、单次攻击伤害以及攻击范围。

接下来 s 行，每行是 2 个用空格隔开的整数 x、y，描述了一个炮塔的位置。当然，蚂蚁窝的洞口以及蛋糕所在的位置上一定没有炮塔。

最后一行是一个正整数 t，表示我们模拟游戏的前 t 秒钟。

输出：

如果在第 t 秒或之前蚂蚁抢到了蛋糕，输出一行“Game over after x seconds”，其中 x 为游戏结束的时间，否则输出“The game is going on”。

如果游戏在 t 秒或之前结束，输出游戏结束时所有蚂蚁的信息，否则输出 t 秒后所有蚂蚁的信息。格式如下：

第一行是 1 个整数 s，表示此时活着的蚂蚁的总数。

接下来 s 行，每行 5 个整数，依次表示一只蚂蚁的年龄（单位为秒）、等级、当前血量，以及在地图上的位置（a，b）。输出按蚂蚁的年龄递减排序。

输入样例：

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

输出样例：

```
The game is going on
5
5 1 3 1 4
4 1 3 0 4
3 1 3 0 3
2 1 3 0 2
1 1 4 0 1
```

样例说明：

3*5 的地图，有 1 个单次伤害为 1、攻击范围为 2 的激光炮塔，它的位置为（2，2），模拟游戏的前 5 秒。5 秒内有 5 只蚂蚁出生，都是向东爬行，其中第 1~4 只在路过（0，2）点时被激光塔伤了 1 格血。在第 5 秒的时候，最早出生的蚂蚁按移动规则 1~3 本来该向东移动，但由于规则 4 的作用，它在发现向北和向西移动都会到达不可达点后，最终选择了向南移动。

数据说明：

100%的数据满足 $1 \leqslant n,m \leqslant 8$，$s \leqslant 20$，$t \leqslant 200,000$