

作业8：魔兽世界

【作业概述】

通过编制一个流行游戏的简化版，重点练习类的组织、设计和继承等技术，实现基于时间轴的事件输出（输出内容相当于游戏流程与结果）。

【背景介绍】

魔兽世界的西面是红魔军的司令部，东面是蓝魔军的司令部。两个司令部之间是依次排列的若干城市，城市从西向东依次编号为1, 2, 3...N($N \leq 20$)。两军的司令部都会制造武士。武士一共有dragon、ninja、iceman、lion、wolf五种。每种武士都有编号、生命值、攻击力这三种属性。双方的武士编号都是从1开始计算。红方制造出来的第n个武士，编号就是n。同样，蓝方制造出来的第n个武士，编号也是n。武士在刚降生的时候有一个初始的生命值，生命值在战斗中会发生变化，如果生命值减少到0或小于0，则武士死亡（消失）。

- 在每个整点，即每个小时的第0分：双方的司令部中各有一个武士降生。
 - 红方司令部按照iceman、lion、wolf、ninja、dragon的顺序制造武士。
 - 蓝方司令部按照lion、dragon、ninja、iceman、wolf的顺序制造武士。
 - 制造武士需要生命元。制造一个初始生命值为m的武士，司令部中的生命元就要减少m个。
 - 如果司令部中的生命元不足以制造某武士，那么司令部就等待，直到获得足够生命元后的第一个整点，才制造该武士。例如，在2:00，红方司令部本该制造一个wolf，如果此时生命元不足，那么就会等待，直到生命元足够后的下一个整点，才制造一个wolf。
- 在每个小时的第10分：所有的武士朝敌人司令部方向前进一步。即从己方司令部走到相邻城市，或从一个城市走到下一个城市，或从和敌军司令部相邻的城市到达敌军司令部。
 - 武士到达对方司令部后就算完成任务了，从此就呆在那里无所事事。
 - 任何一方的司令部里若是出现了2个敌人，则认为该司令部已被敌人占领。
 - 任何一方的司令部被敌人占领，则战争结束。战争结束之后就不会发生任何事情了。
- 在每个小时的第20分：每个城市产出10个生命元。生命元留在城市，直到被武士取走。
- 在每个小时的第30分：如果某个城市中只有一个武士，那么该武士取走该城市中的所有生命元，并立即将这些生命元传送到其所属的司令部。
- 在每个小时的第40分：在有二个武士的城市（注意不包括司令部），会发生战斗。
 - 每次战斗有且只有一方武士主动进攻。被攻击者生命值会减去进攻者的攻击力值。如果被攻击者生命值减至小于等于0，则其被杀死。
 - 在插红旗的城市，以及编号为奇数的无旗城市，由红武士主动发起进攻。
 - 在插蓝旗的城市，以及编号为偶数的无旗城市，由蓝武士主动发起进攻。
 - 如果被攻击者没有被杀死，则会发动反击，被反击者的生命值要减去反击者攻击力值的一半(去尾取整)。反击当然可能致敌人于死地。
 - 如果武士在战斗中杀死敌人（不论是主动进攻杀死还是反击杀死），则其司令部会立即向其发送8个生命元作为奖励，使其生命值增加8。当然前提是司令部得有8个生命元。如果司令部的生命元不足以奖励所有的武士，则优先奖励距离敌方司令部近的武士。
 - 如果某武士在某城市的战斗中杀死了敌人，则该武士的司令部立即取得该城市中所有的生命元。注意，司令部总是先奖励武士，然后再收回打了胜仗的城市的生命元。对于因司令部生命元不足而领不到奖励的武士，司令部也不会取得战利品生命元后为其补发奖励。
 - 每次战斗都只有一方主动进攻一次，反击（如有）也仅限一次。战斗并非是你死我活的，可能是双方都幸存。如果双方都幸存，则双方都不能拿走发生战斗的城市的生命元。

- 一开始所有的城市都是无旗的。当某个城市有连续两场战斗都是同一方的武士杀死敌人(两场战斗中如果有若干个战斗时刻并没有发生战斗, 则这两场战斗仍然算是连续的), 那么该城市就会插上胜方的旗帜, 若原来插着败方的旗帜, 则败方旗帜落下。旗帜一旦插上, 就一直插着, 直到被敌人更换。一个城市最多只能插一面旗帜, 旗帜没被敌人更换前, 也不会再次插同颜色的旗。
- 不同的武士各有其特点:
 - dragon在一次在它主动进攻的战斗结束后, 如果还没有战死, 就会欢呼。
 - ninja挨打了也从不反击敌人。
 - iceman每前进两步, 在第2步完成的时候, 生命值会减少9, 攻击力会增加20。但是若生命值减9后会小于等于0, 则生命值不减9,而是变为1。即iceman不会因走多了而死。
 - lion若是战死, 则其战斗前的生命值就会转移到对手身上。
 - 在一个wolf通过主动攻击杀死敌人的次数达到偶数的时刻(次数从1开始算), 在战斗完成后, 该wolf生命值和攻击力都增加1倍。如果其杀死的敌人是lion,则攻击力和生命值先加倍, 然后才吸取lion的生命值。获取总部的生命元奖励, 也发生在加倍之后。只有在主动攻击杀死敌人时才能有加倍奖励, 反击杀死敌人则不会发生加倍。
- 6. 在每个小时的第50分: 司令部报告它拥有的生命元数量。

【功能要求和范例】

给定一个时间, 要求你将从0点0分开始到此时间为止的所有事件按顺序输出。

事件输出样例

1. 武士降生
输出样例: `000:00 blue lion 1 born`
表示在0点0分, 编号为1的蓝魔lion武士降生
2. 武士前进到某一城市
输出样例: `000:10 red iceman 1 marched to city 1 with 20 elements and force 30`
表示在0点10分, 红魔1号武士iceman前进到1号城市, 此时他生命值为20,攻击力为30
对于iceman,输出的生命值和攻击力应该是变化后的数值
3. 武士主动进攻
输出样例: `000:40 red iceman 1 attacked blue lion 1 in city 1 with 20 elements and force 30`
表示在0点40分, 1号城市中, 红魔1号武士iceman进攻蓝魔1号武士lion,在发起进攻前, 红魔1号武士iceman生命值为20, 攻击力为30
4. 武士反击
输出样例: `001:40 blue dragon 2 fought back against red lion 2 in city 1`
表示在1点40分, 1号城市中, 蓝魔2号武士dragon反击红魔2号武士lion
5. 武士战死
输出样例: `001:40 red lion 2 was killed in city 1`
6. 武士欢呼
输出样例: `003:40 blue dragon 2 yelled in city 4`
7. 武士获取生命元(elements)
输出样例: `001:40 blue dragon 2 earned 10 elements for his headquarter`
8. 旗帜升起
输出样例: `004:40 blue flag raised in city 4`
9. 武士抵达敌军司令部
输出样例: `001:10 red iceman 1 reached blue headquarter with 20 elements and force 30`
(此时他生命值为20,攻击力为30) 对于iceman,输出的生命值和攻击力应该是变化后的数值
10. 司令部被占领
输出样例: `003:10 blue headquarter was taken`
11. 司令部报告生命元数量
输出样例:
`000:50 100 elements in red headquarter`

```
000:50 120 elements in blue headquarter
```

表示在0点50分，红方司令部有100个生命元，蓝方有120个

事件输出提示

- 首先按时间顺序输出；
- 同一时间发生的事件，按发生地点从西向东依次输出。武士前进的事件，算是发生在目的地。
- 在一次战斗中有可能发生上面的3至8号事件。这些事件都算同时发生，其时间就是战斗开始时间。一次战斗中的这些事件，序号小的应该先输出。
- 两个武士同时抵达同一城市，则先输出红武士的前进事件，后输出蓝武士的。
- 显然，10号事件发生之前的一瞬间一定发生了9号事件。输出时，这两件事算同一时间发生，但是应先输出9号事件。
- 虽然任何一方的司令部被占领之后，就不会有任何事情发生了。但和司令部被占领同时发生的事件，全都要输出。

输入格式

输入为一个文本文件。第一行是一个整数,代表测试数据组数 每组测试数据共三行。第一行，三个整数M，N，T。其含义为：每个司令部一开始都有M个生命元($1 \leq M \leq 1000$) 两个司令部之间一共有N个城市($1 \leq N \leq 20$) 要求输出从0时0分开始，到时间T为止(包括T)的所有事件。T以分钟为单位， $0 \leq T \leq 1000$ 第二行：五个整数，依次是dragon、ninja、iceman、lion、wolf的初始生命值。它们都大于0小于等于100 第三行：五个整数，依次是dragon、ninja、iceman、lion、wolf的攻击力。它们都大于0小于等于100

输出格式

本题目要求使用文件输出。对每组测试数据，首先输出“Case:n”，n是测试数据的编号，从1开始，各个Case之间使用一个空行分隔。接下来按恰当的顺序和格式输出到时间T为止发生的所有事件。每个事件都以事件发生的时间开头，时间格式是“时:分”，“时”有三位，“分”有两位，每个事件占一行。详见下附范例

参考范例

输入：

```
2
99 2 1000
10 20 50 50 30
20 50 50 50 50
40 1 1000
20 20 20 20 20
20 20 20 20 20
```

输出：

```
Case:1
000:00 red iceman 1 born
000:00 blue lion 1 born
000:10 red iceman 1 marched to city 1 with 50 elements and force 50
000:10 blue lion 1 marched to city 2 with 50 elements and force 50
000:30 red iceman 1 earned 10 elements for his headquarter
000:30 blue lion 1 earned 10 elements for his headquarter
000:50 59 elements in red headquarter
000:50 59 elements in blue headquarter
001:00 red lion 2 born
001:00 blue dragon 2 born
001:10 red lion 2 marched to city 1 with 50 elements and force 50
001:10 blue lion 1 marched to city 1 with 50 elements and force 50
```

```
001:10 red iceman 1 marched to city 2 with 41 elements and force 70
001:10 blue dragon 2 marched to city 2 with 10 elements and force 20
001:40 red lion 2 attacked blue lion 1 in city 1 with 50 elements and force 50
001:40 blue lion 1 was killed in city 1
001:40 red lion 2 earned 10 elements for his headquarter
001:40 blue dragon 2 attacked red iceman 1 in city 2 with 10 elements and force 20
001:40 red iceman 1 fought back against blue dragon 2 in city 2
001:40 blue dragon 2 was killed in city 2
001:40 red iceman 1 earned 10 elements for his headquarter
001:50 21 elements in red headquarter
001:50 49 elements in blue headquarter
002:00 blue ninja 3 born
002:10 red lion 2 marched to city 2 with 100 elements and force 50
002:10 blue ninja 3 marched to city 2 with 20 elements and force 50
002:10 red iceman 1 reached blue headquarter with 29 elements and force 70
002:40 blue ninja 3 attacked red lion 2 in city 2 with 20 elements and force 50
002:40 red lion 2 fought back against blue ninja 3 in city 2
002:40 blue ninja 3 was killed in city 2
002:40 red lion 2 earned 10 elements for his headquarter
002:40 red flag raised in city 2
002:50 23 elements in red headquarter
002:50 29 elements in blue headquarter
003:10 red lion 2 reached blue headquarter with 58 elements and force 50
003:10 blue headquarter was taken
```

Case:2

```
000:00 red iceman 1 born
000:00 blue lion 1 born
000:10 red iceman 1 marched to city 1 with 20 elements and force 20
000:10 blue lion 1 marched to city 1 with 20 elements and force 20
000:40 red iceman 1 attacked blue lion 1 in city 1 with 20 elements and force 20
000:40 blue lion 1 was killed in city 1
000:40 red iceman 1 earned 10 elements for his headquarter
000:50 22 elements in red headquarter
000:50 20 elements in blue headquarter
001:00 red lion 2 born
001:00 blue dragon 2 born
001:10 red lion 2 marched to city 1 with 20 elements and force 20
001:10 blue dragon 2 marched to city 1 with 20 elements and force 20
001:10 red iceman 1 reached blue headquarter with 39 elements and force 40
001:40 red lion 2 attacked blue dragon 2 in city 1 with 20 elements and force 20
001:40 blue dragon 2 was killed in city 1
001:40 red lion 2 earned 10 elements for his headquarter
001:40 red flag raised in city 1
001:50 12 elements in red headquarter
001:50 0 elements in blue headquarter
002:10 red lion 2 reached blue headquarter with 20 elements and force 20
002:10 blue headquarter was taken
```

【作业要求】

1. 本题目应当应用OO方法编程，所有涉及的对象均应当实现为对应的类，例如司令部、城市、武士，除数据属性外，对象应当支持必要的操作，如事件报告、战争等。
2. 建议实体类派生于共同基类，比如城市和司令部可以派生于虚拟的城市类（场景），武士派生于共同的动物类（精灵）。各派生类的通用动作（共同的动作）可以用基类指针动态调用虚函数的方式实现，这样流程界面代码就会得到很大的简化，即可以通过一个遍历完成一个时刻所有对象的操作。
3. 时间轴（tickle）可以用一个循环实现，其中各个对象依次根据时间轴进度实施动作，并报告状态或事件，循环以某个总部沦陷作为退出点。

【Bonus】

提供图形界面（需使用C++实现，例如Qt、GTK、MFC等）