编程作业: 编程作业—多态与虚函数

您还未提交解答。您必须获得 80/100 分才能通过。

截止时间 在以下日期前通过此作业 五月 14, 11:59 晚上 PDT

> **说明**
>
> 我提交的作业
>
> 讨论

准备

在开始下面的作业前，请先点击这里下载代码模版。

编程题 #1

来源: POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意： 总时间限制: 1000ms 内存限制: 65536kB

下面程序的输出结果是：

A::Fun

C::Do

请填空：

```
#include <iostream>
using namespace std;
class A {
    private:
        int nVal;
    public:
        void Fun()
        { cout << "A::Fun" << endl; }
        void Do()
        { cout << "A::Do" << endl; }
};
class B:public A {
    public:
        virtual void Do()
        { cout << "B::Do" << endl; }
};
class C:public B {
    public:
    void Do( )
    { cout << "C::Do" <<endl; }
    void Fun()
    { cout << "C::Fun" << endl; }
};
void Call(
// 在此处补充你的代码
        ) {
    p.Fun(); p.Do();
}
int main() {
    C c; Call(c);
    return 0;
}
```

输入

无

输出

A::Fun

C::Do

样例输入

```
无
```

样例输出

```
A::Fun
C::Do
```

编程题 # 2

来源: POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意： 总时间限制: 1000ms 内存限制: 65536kB

描述

下面程序的输出结果是：

destructor B

destructor A

请完整写出 class A。 限制条件：不得为 class A 编写构造函数。

```
#include <iostream>
using namespace std;
class A {
// 在此处补充你的代码
};
class B:public A {
    public:
        ~B() { cout << "destructor B" << endl; }
};
int main() {
    A * pa;
    pa = new B;
    delete pa;
    return 0;
}
```

输入

无

输出

destructor B

destructor A

样例输入

```
无
```

样例输出

```
destructor B
destructor A
```

编程题 #3

来源: POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意： 总时间限制: 1000ms 内存限制: 65536kB

描述

下面的程序输出结果是：

A::Fun

A::Do

A::Fun

C::Do

请填空：

```
#include <iostream>
using namespace std;
class A {
    private:
        int nVal;
    public:
        void Fun()
        { cout << "A::Fun" << endl; }
        virtual void Do()
        { cout << "A::Do" << endl; }
};
class B:public A {
    public:
        virtual void Do()
        { cout << "B::Do" << endl; }
};
class C:public B {
    public:
        void Do( )
        { cout << "C::Do" << endl; }
        void Fun()
        { cout << "C::Fun" << endl; }
};
void Call(
// 在此处补充你的代码
        ) {
    p->Fun(); p->Do();
}
int main() {
    Call( new A() );
    Call( new C() );
    return 0;
}
```

输入

无

输出

A::Fun

A::Do

A::Fun

C::Do

样例输入

```
无
```

样例输出

```
A::Fun
A::Do
A::Fun
C::Do
```

编程题 # 4： 魔兽世界终极版

来源: POJ (Coursera声明：在POJ上完成的习题将不会计入Coursera的最后成绩。)

注意： 总时间限制: 2000ms 内存限制: 65536kB

描述

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 <= 20)。红魔军的司令部算作编号为0的城市，蓝魔军的司令部算作编号为N+1的城市。司令部有生命元，用于制造武士。

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（生命值变为负数时应当做变为0处理），则武士死亡（消失）。

有的武士可以拥有武器。武器有三种，sword, bomb,和arrow，编号分别为0,1,2。

武士降生后就朝对方司令部走，在经过的城市如果遇到敌人（同一时刻每个城市最多只可能有1个蓝武士和一个红武士），就会发生战斗。每次战斗只有一方发起主动进攻一次。被攻击者生命值会减去进攻者的攻击力值和进攻者手中sword的攻击力值。被进攻者若没死，就会发起反击，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)和反击者手中sword的攻击力值。反击可能致敌人于死地。

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，司令部总是先完成全部奖励工作，然后才开始从各个打了胜仗的城市回收生命元。对于因司令部生命元不足而领不到奖励的武士，司令部也不会在取得战利品生命元后为其补发奖励。

如果一次战斗的结果是双方都幸存(平局)，则双方都不能拿走发生战斗的城市的生命元。

城市可以插旗子，一开始所有城市都没有旗子。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。

当某个城市有连续两场战斗都是同一方的武士杀死敌人(两场战斗之间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；但如果中间有平局的战斗，就不算连续了)，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。

各种武器有其特点：

sword武器的初始攻击力为拥有它的武士的攻击力的20%（去尾取整）。但是sword每经过一次战斗(不论是主动攻击还是反击)，就会变钝，攻击力变为本次战斗前的80% (去尾取整)。sword攻击力变为0时，视为武士失去了sword。如果武士降生时得到了一个初始攻击力为0的sword，则视为武士没有sword.

arrow有一个攻击力值R。如果下一步要走到的城市有敌人，那么拥有arrow的武士就会放箭攻击下一个城市的敌人（不能攻击对方司令部里的敌人）而不被还击。arrow使敌人的生命值减少R，若减至小于等于0，则敌人被杀死。arrow使用3次后即被耗尽，武士失去arrow。两个相邻的武士可能同时放箭把对方射死。

拥有bomb的武士，在战斗开始前如果判断自己将被杀死（不论主动攻击敌人，或者被敌人主动攻击都可能导致自己被杀死，而且假设武士可以知道敌人的攻击力和生命值），那么就会使用bomb和敌人同归于尽。武士不预测对方是否会使用bomb。

武士使用bomb和敌人同归于尽的情况下，不算是一场战斗，双方都不能拿走城市的生命元，也不影响城市的旗帜。

不同的武士有不同的特点。

dragon可以拥有一件武器。编号为n的dragon降生时即获得编号为 n%3 的武器。dragon还有“士气”这个属性，是个浮点数，其值为它降生后其司令部剩余生命元的数量除以造dragon所需的生命元数量。dragon 在一次在它主动进攻的战斗结束后，如果还没有战死，而且士气值大于0.8，就会欢呼。dragon每取得一次战斗的胜利(敌人被杀死)，士气就会增加0.2，每经历一次未能获胜的战斗，士气值就会减少0.2。士气增减发生在欢呼之前。

ninjia可以拥有两件武器。编号为n的ninjia降生时即获得编号为 n%3 和 (n+1)%3的武器。ninja 挨打了也从不反击敌人。

iceman有一件武器。编号为n的iceman降生时即获得编号为 n%3 的武器。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 有“忠诚度”这个属性，其初始值等于它降生之后其司令部剩余生命元的数目。每经过一场未能杀死敌人的战斗，忠诚度就降低K。忠诚度降至0或0以下，则该lion逃离战场,永远消失。但是已经到达敌人司令部的lion不会逃跑。Lion在己方司令部可能逃跑。lion 若是战死，则其战斗前的生命值就会转移到对手身上。所谓“战斗前”，就是每个小时的40分前的一瞬间。

wolf降生时没有武器，但是在战斗中如果获胜（杀死敌人），就会缴获敌人的武器，但自己已有的武器就不缴获了。被缴获的武器当然不能算新的，已经被用到什么样了，就是什么样的。

以下是不同时间会发生的不同事件：

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5分，该逃跑的lion就在这一时刻逃跑了。

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。

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35分，拥有arrow的武士放箭，对敌人造成伤害。放箭事件应算发生在箭发出的城市。注意，放箭不算是战斗，因此放箭的武士不会得到任何好处。武士在没有敌人的城市被箭射死也不影响其所在城市的旗帜更换情况。

在每个小时的第38分，拥有bomb的武士评估是否应该使用bomb。如果是，就用bomb和敌人同归于尽。

在每个小时的第40分：在有两个武士的城市，会发生战斗。 如果敌人在5分钟前已经被飞来的arrow射死，那么仍然视为发生了一场战斗，而且存活者视为获得了战斗的胜利。此情况下不会有“武士主动攻击”，“武士反击”，“武士战死”的事件发生，但战斗胜利后应该发生的事情都会发生。如Wolf一样能缴获武器，旗帜也可能更换，等等。在此情况下,Dragon同样会通过判断是否应该轮到自己主动攻击来决定是否欢呼。

在每个小时的第50分，司令部报告它拥有的生命元数量。

在每个小时的第55分，每个武士报告其拥有的武器情况。

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。

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。事件及其对应的输出样例如下：

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

如果造出的是dragon，那么还要多输出一行，例：

000:00 blue dragon 1 born

Its morale is 23.34

表示该该dragon降生时士气是23. 34(四舍五入到小数点后两位)

如果造出的是lion，那么还要多输出一行，例:

000:00 blue lion 1 born

Its loyalty is 24

表示该lion降生时的忠诚度是24

2) lion逃跑

输出样例： 000:05 blue lion 1 ran away

表示在 0点5分，编号为1的蓝魔lion武士逃走

3) 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

4)武士放箭

输出样例： 000:35 blue dragon 1 shot

表示在 0点35分，编号为1的蓝魔dragon武士射出一支箭。如果射出的箭杀死了敌人，则应如下输出：

000:35 blue dragon 1 shot and killed red lion 4

表示在 0点35分，编号为1的蓝魔dragon武士射出一支箭，杀死了编号为4的红魔lion。

5)武士使用bomb

输出样例： 000:38 blue dragon 1 used a bomb and killed red lion 7

表示在 0点38分，编号为1的蓝魔dragon武士用炸弹和编号为7的红魔lion同归于尽。

6) 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

7) 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

8) 武士战死

输出样例：001:40 red lion 2 was killed in city 1

被箭射死的武士就不会有这一条输出。

9) 武士欢呼

输出样例：003:40 blue dragon 2 yelled in city 4

10) 武士获取生命元(elements)

输出样例：001:40 blue dragon 2 earned 10 elements for his headquarter

11) 旗帜升起

输出样例：004:40 blue flag raised in city 4

12) 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

13) 司令部被占领

输出样例：003:10 blue headquarter was taken

14)司令部报告生命元数量

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

15)武士报告武器情况

000:55 blue wolf 2 has arrow(2),bomb,sword(23)

000:55 blue wolf 4 has no weapon

000:55 blue wolf 5 has sword(20)

表示在0点55分，蓝魔2号武士wolf有一支arrow（这支arrow还可以用2次），一个bomb，还有一支攻击力为23的sword。

蓝魔4号武士wolf没武器。

蓝魔5号武士wolf有一支攻击力为20的sword。

交代武器情况时，次序依次是：arrow,bomb,sword。如果没有某种武器，某种武器就不用提。报告时，先按从西向东的顺序所有的红武士报告，然后再从西向东所有的蓝武士报告。

输出事件时：

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 6 至 11 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，13号事件发生之前的一瞬间一定发生了12号事件。输出时，这两件事算同一时间发生，但是应先输出12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

第一行是t,代表测试数据组数

每组样例共三行。

第一行，五个整数 M,N,R,K, T。其含义为：

每个司令部一开始都有M个生命元(1 <= M <= 10000)

两个司令部之间一共有N个城市(1 <= N <= 20)

arrow的攻击力是R

lion每经过一场未能杀死敌人的战斗，忠诚度就降低K。

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 <= T <= 5000

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10000

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10000

输出

对每组数据，先输出一行：

Case n:

如对第一组数据就输出 Case1:

然后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。

样例输入

```
1
20 1 10 10 1000
20 20 30 10 20
5 5 5 5 5
```

样例输出

```
Case 1:
000:00 blue lion 1 born
Its loyalty is 10
000:10 blue lion 1 marched to city 1 with 10 elements and force 5
000:30 blue lion 1 earned 10 elements for his headquarter
000:50 20 elements in red headquarter
000:50 20 elements in blue headquarter
000:55 blue lion 1 has no weapon
001:00 blue dragon 2 born
Its morale is 0.00001:10 blue lion 1 reached red headquarter with 10 elements and force 5
001:10 blue dragon 2 marched to city 1 with 20 elements and force 5
001:30 blue dragon 2 earned 10 elements for his headquarter
001:50 20 elements in red headquarter
001:50 10 elements in blue headquarter
001:55 blue lion 1 has no weapon
001:55 blue dragon 2 has arrow(3)
002:10 blue dragon 2 reached red headquarter with 20 elements and force 5
002:10 red headquarter was taken
```