10802 CPP Midterm Exam

Contributor：Pei-Ru Wang

Subject：Tactical role-playing game (TRPG)

Main testing concept：

Basics

■ C++ BASICS
■ FLOW OF CONTROL
■ FUNCTION BASICS
□ PARAMETERS AND OVERLOADING
□ ARRAYS
■ STRUCTURES AND CLASSES
□ CONSTRUCTORS AND OTHER TOOLS
□ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES
□ STRINGS
□ POINTERS AND DYNAMIC ARRAYS

Functions

□ SEPARATE COMPILATION AND NAMESPACES
□ STREAMS AND FILE I/O
□ RECURSION
□ INHERITANCE
□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ TEMPLATES
□ LINKED DATA STRUCTURES
□ EXCEPTION HANDLING
□ STANDARD TEMPLATE LIBRARY
□ PATTERNS AND UML

Description：

Please design a game where the player is imprisoned on an island full of monsters and the player needs to fight against these monsters until the player kills all of monsters to gain victory.
Each character has three attributes: **hp**, **attack**, and **defense**. If the **hp** is less than or equal to 0, the character will die. The attack value of each character equals the **attack** minus the enemy's **defense** (If the calculated attack value is less than 0, the enemy will not be harmed). For equality, only one monster will fight the player in each round. The player's hp will not recover after each round, and he/she need to fight the monsters one by one until all monsters die or he/she die.
During the battle, the player and the monster take turns to attack until one side is killed.

Input：

The first line has three integers to list out the attributes of the hero, and all the following other lines also have three integers to list out the attributes of monsters.
Note that

1. Each line contains three integers: **hp**, **attack**, and **defense**.
2. Fighting sequence is defined by the input order of monster enemies.
3. The player attacks first in each round.

** The number of monsters is variable, keep reading inputs until EOF is read.
** The input parameters will all be legal and you won't need to do any error check on the input.

Output：

1. At the end of each round, please output the status of the player and fighting monster by listing the hp of them in the format of "Player: <**player_hp**> Enemy: <**monster_hp**>".
2. Please output each fighting result. While the player kills the fighting monster, you should output "**Player Win**", and the game continues to let the player fight against the next monster

until all the monsters are killed. While the player loses, you should output "**Player Dead**", and the game is over.

Sample Input / Output :

Sample Input	Sample Output
100 12 9 20 8 3	Player:100 Enemy:11 Player:100 Enemy:11 Player:100 Enemy:2 Player:100 Enemy:2 Player:100 Enemy:-7 Player Win

40 12 5
20 10 3
30 15 4
	Player:40 Enemy:11
	Player:35 Enemy:11
	Player:35 Enemy:2
	Player:30 Enemy:2
	Player:30 Enemy:-7
	Player Win
	Player:30 Enemy:22
	Player:20 Enemy:22
	Player:20 Enemy:14
	Player:10 Enemy:14
	Player:10 Enemy:6
	Player:0 Enemy:6
	Player Dead

40 12 5
10 40 8
5 10 7
	Player:40 Enemy:6
	Player:5 Enemy:6
	Player:5 Enemy:2
	Player:-30 Enemy:2
	Player Dead

■ **Easy, only basic programming syntax and structure are required.**

□ **Medium, multiple programming grammars and structures are required.**

□ **Hard, need to use multiple program structures or complex data types.**

Expected solving time:

10 minutes

Other notes: