

Counter Strike

Intro

Your task is to write a program that represents parts from the famous computer game – “**Counter Strike**”. You will be position right in the middle of a round between the Terrorist and the Counter-Terrorists. Usually in that game the very first round of the game is called a “Pistol Round” and it very important. In that round both playing teams have money only to purchase guns. The winner of this round has enough money to buy automatic weapons in the next round, while the loser team sticks to their pistols.

Provided input

You are provided with folder called **Skeleton**, which contains:

- A header file that contains the most important information for the task.
- A Basic Skeleton to guide you in the initial direction. You don't need to strictly follow it.

You are also provided with folder called **tests**, which contains all of the tests that your program should pass.

- There are a total of 13 tests. 13 inputs and 13 outputs. The 3 of the tests are copy of the ones described in this document. The other 10 are unique.

Task description

Your task is to provide to **provide an implementation**, which with the given standard input will produce the correct standard output (detailed description below).

The game is played with two players from opposing teams (Player 0 and Player 1).

In game you should find a header prototypes for the 2 different supported Pistol types:

```
enum PistolType {  
    GLOCK, DESERT_EAGLE  
};
```

You must provide concrete implementations for the DesertEagle and Glock pistols.

In the game both players takes turns shooting each other with their provided pistols for the round. First goes Player 0, then Player 1 ... and so on until one of them dies (has no health left).

Each player has 2 vital attributes “health” and “armor”

```
typedef struct {  
    int health;  
    int armor;  
} PlayerVitalData;
```

The armor is a way for the player to negate some of the incoming damage that he is about to take. A player will not always have armor.

Each player also acquires a Pistol at the start of the round (determined by the input). This Pistol is either a DesertEagle or a Glock. Both Pistols have the same attributes, but they do have different effects.

```
typedef struct {  
    enum PistolType pistolType;  
    int damagePerRound;  
    int clipSize;  
    int currClipBullets;  
    int remainingAmmo;  
} Pistol;
```

The Pistol attributes explained:

pistolType – which pistol the player is wielding

damagePerRound – indicates how much damage does each bullet deal

clipSize – indicates how much bullet capacity the concrete pistol clip has

currClipBullets – indicates how many bullets are left in the current clip

remainingAmmo – indicates how many spare bullets are left for the player to reload with.

This count does not include in itself the bullets that are loaded in the current pistol clip

The input to the program consists of (in that order):

player 0 health, player 0 armor

player 1 health, player 1 armor

player 0 PistolType, player 0 pistol damage per round, player 0 clipSize, player 0 remainingAmmo

player 1 PistolType, player 1 pistol damage per round, player 1 clipSize, player 1 remainingAmmo

Game rules:

- At the start of the game – each player has a full clip of bullets ready and loaded.
- Players take turns shooting at each other (starting always from Player with ID: 0)
- If a player pistol is required to fire and current bullets in the **clip reaches 0** – the player **must reload its pistol. The player loses his remaining turn and does not deal any more damage for this turn.**

Common **DesertEagle** and **Glock** rules:

- If an opponent has a **positive armor value** and the shot fired against this opponent has a **bigger value than the armor** – the armor is set to 0 and the remaining damage is **subtracted** from the target's health indicator.
- Both **DesertEagle** and **Glock** ::fire() methods should return a bool variable, which indicates whether or not the target (enemy player) has been killed by this round bullets that were fired.
An opponent is considered killed when his health indicator drops below 0 (or is equal to 0).

DesertEagle Pistol specifics (see concrete examples at the 'Examples' section):

- If the opponent has no armor:
 - Deals 100% of the pistol original damage to the opponent's health indicator
- If the opponent has armor:
 - Deals 75% of the pistol original damage to the opponent's health indicator
 - Deals 25% of the pistol original damage to the opponent's armor indicator
- Pistol damage will **always** be dividable by 4 (Example: 32, 16, 4, 100)
- When reloading – no shots will be made in the same turn.

Glock Pistol specifics (see concrete examples at the ‘Examples’ section):

- **Fires 3 bullets one after another for a single turn.** A print to the console should be made **for each** shot fired.
If there are less than 3 bullets remaining for this turn – fire all the remaining bullets and initiate a **reload**. No bullets should be fired after the reload in the same turn.
- If the opponent has no armor:
 - Deals 100% of the pistol original damage to the opponent’s health indicator
- If the opponent has armor:
 - Deals 50% of the pistol original damage to the opponent’s health indicator
 - Deals 50% of the pistol original damage to the opponent’s armor indicator
- Pistol damage will **always** be dividable by 4 (Example: 36, 8, 112, 44)

Console output

- After each shot fired you should **print** to the console the opponent player vital details following this exact format:
"Enemy left with: **N** health and **M** armor" followed by a **newline**.
Where **N** indicates the remaining opponents **health** (after the shot) and **M** indicates the remaining opponents **armor** (after the shot)
- If a player has no bullets left on each of his next turns print “No ammo left” followed by a **newline**.

Restrictions

Code should compile under the C 11 standard.

Code should not use any non-standard extensions.

Examples

Input	Output
100 0 120 0 0 8 9 71 1 24 7 35	PlayerID 0 turn: Enemy left with: 112 health and 0 armor Enemy left with: 104 health and 0 armor Enemy left with: 96 health and 0 armor PlayerID 1 turn: Enemy left with: 76 health and 0 armor PlayerID 0 turn: Enemy left with: 88 health and 0 armor Enemy left with: 80 health and 0 armor Enemy left with: 72 health and 0 armor PlayerID 1 turn:

Enemy left with: 52 health and 0 armor

PlayerID 0 turn:

Enemy left with: 64 health and 0 armor

Enemy left with: 56 health and 0 armor

Enemy left with: 48 health and 0 armor

PlayerID 1 turn:

Enemy left with: 28 health and 0 armor

PlayerID 0 turn:

Reloading...

currClipBullets: 9, remainingAmmo: 62

PlayerID 1 turn:

Enemy left with: 4 health and 0 armor

PlayerID 0 turn:

Enemy left with: 40 health and 0 armor

Enemy left with: 32 health and 0 armor

Enemy left with: 24 health and 0 armor

PlayerID 1 turn:

Enemy left with: -20 health and 0 armor

Player with ID: 1 wins!

100 50
120 50
0 12 8 50
1 24 4 24

PlayerID 0 turn:

Enemy left with: 114 health and 44 armor

Enemy left with: 108 health and 38 armor

Enemy left with: 102 health and 32 armor

PlayerID 1 turn:

Enemy left with: 82 health and 44 armor

PlayerID 0 turn:

Enemy left with: 96 health and 26 armor

Enemy left with: 90 health and 20 armor

Enemy left with: 84 health and 14 armor

PlayerID 1 turn:

Enemy left with: 64 health and 38 armor

PlayerID 0 turn:

Enemy left with: 78 health and 8 armor

Enemy left with: 72 health and 2 armor

Reloading...

currClipBullets: 8, remainingAmmo: 42

PlayerID 1 turn:

Enemy left with: 46 health and 32 armor

PlayerID 0 turn:

Enemy left with: 62 health and 0 armor

Enemy left with: 50 health and 0 armor

Enemy left with: 38 health and 0 armor

PlayerID 1 turn:

Enemy left with: 28 health and 26 armor

PlayerID 0 turn:

Enemy left with: 26 health and 0 armor

Enemy left with: 14 health and 0 armor

Enemy left with: 2 health and 0 armor

PlayerID 1 turn:

Reloading...

currClipBullets: 4, remainingAmmo: 20

PlayerID 0 turn:

Enemy left with: -10 health and 0 armor

Player with ID: 0 wins!

200 100
120 20
1 16 4 3
1 24 7 35

PlayerID 0 turn:

Enemy left with: 108 health and 16 armor

PlayerID 1 turn:

Enemy left with: 182 health and 94 armor

PlayerID 0 turn:

Enemy left with: 96 health and 12 armor

PlayerID 1 turn:

Enemy left with: 164 health and 88 armor

PlayerID 0 turn:

Enemy left with: 84 health and 8 armor

PlayerID 1 turn:

Enemy left with: 146 health and 82 armor

PlayerID 0 turn:

Enemy left with: 72 health and 4 armor

PlayerID 1 turn:

Enemy left with: 128 health and 76 armor

PlayerID 0 turn:

Reloading...

currClipBullets: 3, remainingAmmo: 0

PlayerID 1 turn:

Enemy left with: 110 health and 70 armor

PlayerID 0 turn:

Enemy left with: 60 health and 0 armor

PlayerID 1 turn:

Enemy left with: 92 health and 64 armor

PlayerID 0 turn:

Enemy left with: 44 health and 0 armor

PlayerID 1 turn:

Enemy left with: 74 health and 58 armor

PlayerID 0 turn:

Enemy left with: 28 health and 0 armor

PlayerID 1 turn:

Reloading...

currClipBullets: 7, remainingAmmo: 28

PlayerID 0 turn:

No ammo left

PlayerID 1 turn:

Enemy left with: 56 health and 52 armor

PlayerID 0 turn:

No ammo left

PlayerID 1 turn:

Enemy left with: 38 health and 46 armor

PlayerID 0 turn:

No ammo left

PlayerID 1 turn:

Enemy left with: 20 health and 40 armor

PlayerID 0 turn:

No ammo left

PlayerID 1 turn:

Enemy left with: 2 health and 34 armor

PlayerID 0 turn:

No ammo left

PlayerID 1 turn:

Enemy left with: -16 health and 28 armor

Player with ID: 1 wins!