

---

## Задача: Перестрелка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды на 10 ходов
Ограничение по памяти:	64 мегабайта

Много лет идёт война между Красными и Синими. Сначала они воевали при помощи палок и камней, потом в ход пошли мечи и щиты, потом ружья. И вот теперь основной боевой единицей в этой бессмысленной и беспощадной войне является танк. Каждый танк экипирован энергетической пушкой, стреляющей сгустками энергии и батареей, которая может хранить до 100 единиц энергии. В течение боя экипаж танка может выбирать, какое количество энергии потратить на выстрел. К сожалению, конструкция танка такова, что через 10 выстрелов пушка перегревается и бой на этом заканчивается.

Вам предлагается написать программу управления танком, выбирающую, какое количество энергии требуется потратить на каждый выстрел. При перестрелке двух танков, они одновременно делают 10 выстрелов. Выстрелы двух танков сталкиваются в воздухе. При столкновении выстрел большей мощности нейтрализует менее мощный, после чего продолжает своё движение. Когда выстрел попадает в танк, он наносит танку одну единицу урона (вне зависимости от мощности выстрела). Если сталкиваются два выстрела одинаковой мощности, они взаимоуничтожаются. Вашей задачей будет нанести вражескому танку как можно большее повреждение за 10 выстрелов. Помните, что у вас всего 100 единиц энергии и расходуйте её экономно.

### Формат входного файла

Перед каждым ходом в стандартный поток ввода Вашей программе будет передано состояние предыдущего выстрела, описываемое двумя числами: количеством энергии, которое потратил Ваш танк на выстрел и количеством энергии, которое потратил вражеский танк на выстрел. Завершаться передача состояния будет пустой строкой. Перед первым ходом Вам будут переданы два нуля.

### Формат выходного файла

На каждом ходу Вам требуется вывести количество энергии, которое Вы хотите потратить на очередной выстрел. После вывода хода выведите пустую строку. Не забывайте сбрасывать буфер выходного потока после каждого хода. В C/C++ это можно сделать с помощью `fflush(stdout)`; , в Pascal с помощью `flush(output)`;

### Примеры

стандартный ввод
10 10
стандартный вывод
90

В примере танк расходует всю свою энергию за два хода: 10 единиц на предыдущем ходу и 90 на текущем. Будьте внимательны: Вам не следует завершать свою программу после 10 ходов, система завершит её самостоятельно. Используйте функции чтения для приостановки программы после 10-го хода.

---

Пример реализованной стратегии, на каждом ходу расходуемой случайное количество энергии (C++):

```
#include <iostream>
#include <fstream>
#include <ctime>
#include <cstdlib>

using namespace std;

int main()
{
    srand((unsigned)time(NULL));
    int energy = 100;
    int i;
    int we, them;
    for (i = 0; i < 10; i++)
    {
        scanf("%d%d", &we, &them);
        int cur = rand() % (energy + 1);
        printf("%d\n\n", cur);
        energy -= cur;
        fflush(stdout);
    }
    scanf("%d", &we);
    return 0;
}
```

---

Пример реализованной стратегии, на каждом ходу расходуемой случайное количество энергии (Pascal):

```
var energy : integer;
i, cur, we, them : integer;
begin
randomize;
energy := 100;
for i := 1 to 10 do begin
readln(we, them);
cur := trunc(random() * 1000) mod (energy + 1);
writeln(cur);
writeln;
        flush(output);
        energy := energy - cur;
    end;
    readln(we);
end.
```