

---

## Problem: Shooting

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds for 10 moves  
Memory limit:         64 megabytes

The war between the Red and the Blue is going for a long time. It started with sticks and stones, continued using swords and shield, after that they used rifles. Nowadays the most common unit in this war is a tank. Every tank is equipped with an energy cannon and a battery, which can contain at most 100 units of energy. During the battle tank's crew could choose, how much energy to spend for every shot. Unfortunately, tank is developed in such a way, that after 10 shots the cannon overheats and the battle is finished.

You are to write AI for the tank (nor Red nor Blue want to spend their men in this war). Your program should decide, how much energy to spend for every shot. Battle between two tanks consists of 10 consecutive shots (tanks are shooting simultaneously). Two tanks' shots collide in the air. On of two shots which is more powerful, neutralises the other shot and continues its moving. When the shot reaches the tank it does 1 unit of damage to the tank (regardless of the shot's power). If two shots are equal in amount of energy, they just annihilate in the air and no damage is done to any tank. Your task is to do as much damage to other tank as you can in 10 shots. Remember, that you have only 100 units of energy, so spend it carefully.

### Input

Before every move your program will receive two numbers in standard input: how much energy your and other tank spent for previous shot respectively. Before first move you will receive two zeroes. After these numbers there will be a blank line.

### Output

Output amount of energy you want to spend for the shot. After the number output a blank line. Don't forget to flush the output buffer after move. See sample programs for further clarification.

### Examples

<b>standard input</b>
10 10
<b>standard output</b>
90

In this example the tank uses all energy in two moves: 10 units for previous shot and 90 units for current shot. Be careful: your program should not terminate after 10 moves, the testing system will terminate your program by itself. Use something like reading from stdin to hang after 10-th move.

---

Sample program for 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;
}
```

---

Sample program for 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.
```