Задача: Змейка - 2

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

Многие наверное помнят игру “Змейка”, широко распространённую в китайских игровых консолях наряду с “Тетрисом” и в телефонах одной известной фирмы. В этой задаче Вам предлагается написать программу, играющую в змейку.

Игровое поле представляет собой матрицу клеток размером 30×30. В каждой клетке поля может находиться одна из двух змеек, стена или яблоко. Каждая змейка состоит некоторого количества клеток, пронумерованных, начиная с единицы. На каждом ходу змейка выбирает направление движения, после чего первая клетка змейки сдвигается в этом направлении, а каждая последующая клетка занимает место предыдущей. Если первая клетка змейки оказывается на клетке с яблоком, змейка становится на одну клетку длиннее. Новая клетка добавляется к змейке в конец после сдвига (то есть на ту клетку, где была последняя клетка змейки перед ходом).

Яблоки появляются на поле случайным образом, с частотой примерно 1 яблоко на 10 ходов (здесь ходом считается ход одной змейки и ответ другой).

Изначально каждая змейка состоит из одной клетки, расположенной в клетке $(4, 4)$ или $(25, 25)$ (нумерация клеток с нуля).

Если в результате хода змейки ее голова должна быть сдвинута за границу поля или на клетку, занятую какой-либо змейкой, или на клетку, занятую стеной, ходящая змейка объявляется проигравшей. Если в результате хода змейки её длина достигает 250 клеток, змейка объявляется победившей.

Формат входного файла

Перед каждым ходом в стандартный поток ввода Вашей программе будет передано текущее состояние поля, 30 строк по 30 чисел в каждой. Завершаться передача состояния будет пустой строкой. Каждое из этих чисел a_{ij} соответствует одной клетке поля. Если $0 < a_{ij} \leq 250$, то в этой клетке поля находится клетка Вашей змейки с номером a_{ij}. Если $a_{ij} < 0$, то в этой клетке находится клетка змейки соперника с номером $-a_{ij}$. Если $a_{ij} = 1000$, в клетке находится яблоко. Если $a_{ij} = -1000$, в клетке находится стена. Если же $a_{ij} = 0$, клетка пуста.

Формат выходного файла

На каждом ходу Вам требуется вывести направление, в котором Вы хотите переместить свою змейку. Каждое из четырёх направлений обозначается одним символом: `U` — вверх, `L` — влево, `R` — вправо, `D` — вниз. 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

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 3 2 1 1000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -2 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -3 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -6 -5 -4 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -7 -8 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

стандартный вывод

```
R
```

В примере змейка, совершающая ход, удлинится на одну единицу после хода.

Пример реализованной стратегии, на каждом ходу делающей ход в случайном направлении (C++):

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

using namespace std;

const int SZ = 30;
int field[SZ][SZ];
const char dir[5] = "LURD";

int main()
{
    srand((unsigned)time(NULL));
    int i, j;
    while (1)
    {
        for (i = 0; i < SZ; i++)
        {
            for (j = 0; j < SZ; j++)
            {
                scanf("%d", &field[i][j]);
            }
        }
        printf("%c\n\n", dir[rand() % 4]);
        fflush(stdout);
    }
    return 0;
}
```

Пример реализованной стратегии, на каждом ходу делающей ход в случайную клетку (Pascal):

```
const SZ = 30;
const dir : array[0..3] of char = ('L', 'R', 'U', 'D');
var field : array[1..SZ, 1..SZ] of integer;
i, j : integer;
begin
randomize;
while true do begin
        for i := 1 to SZ do begin
         for j := 1 to SZ do begin
         read(field[i, j]);
         end;
        end;
        writeln(dir[trunc(random() * 1000) mod 4]);
        writeln;
        flush(output);
    end;
end.
```