
Задача: Крестики-нолики

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

В этой задаче Вам предлагается реализовать стратегию игры в крестики-нолики на доске 20×20 .

Напомним правила игры: два игрока ходят по очереди. На своем ходу игрок выбирает свободную клетку поля и ставит туда крестик, если это первый игрок и нолик, если это второй игрок. Выигрывает тот, кто первым расположит пять крестиков (ноликов) в ряд. Если свободных клеток на поле не остается, партия считается сыгранной вничью.

В этой задаче оценивание будет производиться по круговой системе, таким образом Ваше решение будет играть по двадцать партий с каждым другим (десять партий в роли первого игрока, десять — в роли второго). За победу в партии Ваше решение получит 1 очко, за ничью 0.5 очка, за поражение 0 очков.

Формат входного файла

Перед каждым ходом в стандартный поток ввода Вашей программе будет передано текущее состояние поля, 20 строк по 20 чисел в каждой. Завершаться передача состояния будет пустой строкой. Каждое из этих чисел соответствует одной клетке поля и может быть 0, если клетка пуста, 1, если в ней стоит крестик и 2 если в ней стоит нолик.

Определить, играете вы за первого игрока или второго игрока, можно проанализировав самое первое состояние поля, которое Вам будет передано. Если в нём все клетки пустые, то вы — первый игрок, в противном случае — второй.

Формат выходного файла

На каждом ходу Вам требуется вывести номер строки и столбца клетки, в которую Вы хотите сделать ход. Строки поля нумеруются сверху вниз, начиная с нуля. Столбцы нумеруются слева направо, также начиная с нуля. Разделяйте номера строки и столбца пробелом. 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);`

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

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

using namespace std;

const int SZ = 20;
int field[SZ][SZ];

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]);
            }
        }
        while (1)
        {
            int ni, nj;
            ni = rand() % SZ;
            nj = rand() % SZ;
            if (field[ni][nj] == 0)
            {
                printf("%d %d\n\n", ni, nj);
                break;
            }
        }
        fflush(stdout);
    }
    return 0;
}
```

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

```
const SZ = 20;
var field : array[1..SZ, 1..SZ] of integer;
i, j, ni, nj : 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;
        while true do begin
            ni := trunc(random() * 1000) mod SZ + 1;
            nj := trunc(random() * 1000) mod SZ + 1;
            if field[ni][nj] = 0 then begin
                writeln(ni, ' ', nj);
                writeln;
                break;
            end;
        end;
        flush(output);
    end;
end.
```