

Серия G2: Теория игр 2

Во всех задачах следует использовать стандартные потоки ввода и вывода (по умолчанию это клавиатура и экран).

Следующие задачи разбиты на пары.

- В первой задаче каждой пары описана игра для двух игроков. Проигрывает тот, кто не может сделать ход. Если при правильной игре выигрывает первый игрок, следует вывести «First», иначе — «Second».
- Во второй задаче каждой пары требуется играть против программы жюри в описанную игру, делая ходы за первого игрока. Сначала следует прочитать начальные данные об игре.

Чтобы сделать ход, следует вывести на отдельной строке этот ход. Если возможных ходов нет, или участник хочет сдаться заранее, следует вместо хода вывести число -1 . После вывода необходимо очистить буфер вывода: это делается, например, командой `stdout.flush ()`; на языке D или командой `fflush (stdout)`; на языках C и C++.

Далее, если игра окончена, программа может завершить работу. В противном случае следует прочитать ход, сделанный программой жюри, и делать следующий ход. Если вместо хода получено число -1 , это означает победу программы участника.

Решение считается правильным, если оно выигрывает во всех тестах, в которых можно выиграть при правильной игре, а при неизбежном проигрыше рано или поздно выводит -1 .

G2a. Есть один прямоугольный кусок шоколадки из $m \times n$ долек. Ход состоит в том, чтобы взять какой-то из имеющихся кусков шоколадки и разделить его по прямой линии между дольками на два куска, **различных** как прямоугольники.
Заданы два целых числа — m и n ($1 \leq m, n \leq 10$).

G2b. Ход записывается шестью числами: сначала пишутся размеры куска, который следует разделить (два числа), а затем — размеры двух получающихся кусков в любом порядке (дважды по два числа). Порядок чисел в записи размеров одного куска также может быть произвольным.

- G2c.** Есть n вазочек с конфетами: в первой вазочке количество конфет равно a_1 , во второй — a_2, \dots , в n -й — a_n . За один ход можно выбрать одну любую вазочку и взять из неё любое положительное количество конфет.
В первой строке задано одно целое число n ($1 \leq n \leq 100$). Во второй строке заданы числа a_1, a_2, \dots, a_n , разделённые пробелами ($1 \leq a_i \leq 100$).
- G2d.** Ход записывается двумя числами — номером вазочки и количеством конфет, которое следует из неё взять.
- G2e.** То же, что в задаче **G2c**, но за один ход можно взять из одной любой вазочки 1 или 2 конфеты, а не произвольное количество.
- G2f.** То же, что в задаче **G2d**, но за один ход можно взять из одной любой вазочки 1 или 2 конфеты, а не произвольное количество.
- G2g.** То же, что в задаче **G2c**, но за один ход можно взять из одной любой вазочки 2, 3 или 5 конфет, а не произвольное количество.
- G2h.** То же, что в задаче **G2d**, но за один ход можно взять из одной любой вазочки 2, 3 или 5 конфет, а не произвольное количество.
- G2i.** В ряд расположены n палочек: некоторые стоят, а некоторые лежат. За один ход можно уронить либо одну стоящую палочку, либо две соседние стоящие палочки.
В первой строке задано одно целое число n ($1 \leq n \leq 100$). Вторая строка состоит из n символов, каждый из которых — либо «|» (ASCII-код 124, палочка стоит), либо «.» (ASCII-код 46, палочка лежит). Палочки пронумерованы от 1 до n слева направо.
- G2j.** Ход записывается как количество уроненных палочек, после которого следуют номера этих палочек в возрастающем порядке. Палочки пронумерованы числами от 1 до n в изначальном порядке их следования в ряду.
- G2k.** То же, что в задаче **G2i**, но за один ход можно уронить либо две, либо три соседние стоящие палочки.
- G2l.** То же, что в задаче **G2j**, но за один ход можно уронить либо две, либо три соседние стоящие палочки.