

Содержание

Задача 3D. SAT USAT [0.2 sec, 256 mb]

2

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом inc, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Задача 3D. SAT USAT [0.2 sec, 256 mb]

Широко известна задача **3-SAT**. Ещё про неё можно почитать [здесь](#).

Решите её. Гарантируется, что решение существует.

Формулировка **3-SAT**: нужно подобрать значения n булевых переменных так, чтобы все m утверждений вида $x_{i_1} = e_1 \vee x_{i_2} = e_2 \vee x_{i_3} = e_3$ обратились в истину.

Формат входных данных

На первой строке число переменных n и число утверждений m ($1 \leq m \leq \min(n^2, 1000)$).

Каждая из следующих m строк содержит числа $i_1, e_1, i_2, e_2, i_3, e_3$ и задает утверждение $x_{i_1} = e_1 \vee x_{i_2} = e_2 \vee x_{i_3} = e_3$.

Все тесты случайны, тем не менее гарантируется, что решение существует.

Формат выходных данных

Выведите строку из n нулей и единиц — значения переменных.

Если у данной задачи **3-SAT** есть несколько решений, выведите любое.

Система оценки

Подзадача 1 (10 баллов)	$n \leq 20$
Подзадача 2 (40 баллов)	$n \leq 30$
Подзадача 3 (10 баллов)	$n \leq 40$
Подзадача 4 (10 баллов)	$n \leq 50$
Подзадача 5 (10 баллов)	$n \leq 70$
Подзадача 6 (10 баллов)	$n \leq 90$
Подзадача 7 (10 баллов)	$n \leq 120$

Примеры

stdin	stdout
2 3 1 0 1 0 1 0 2 0 2 1 1 1 1 1 2 1 1 1	01

Пояснение к примеру

$$(x_1 = 0 \vee x_1 = 0 \vee x_1 = 0) \wedge (x_2 = 0 \vee x_2 = \underline{1} \vee x_1 = 1) \wedge (x_1 = 1 \vee x_2 = \underline{1} \vee x_1 = 1)$$

В каждом клозе подчеркнуты истинные условия.