

Задача А. Поменяйте одно число

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Это задача с двойным запуском.

Дан массив из пяти целых чисел, каждое — от 1 до 5.

Сначала массив попадает к Маше. Маша выбирает одно из чисел и либо увеличивает его на единицу, либо уменьшает на единицу.

После этого массив неизвестное количество раз поворачивается по циклу. При каждом повороте первое число становится вторым, второе — третьим, и так далее; последнее число становится первым.

Далее массив получает Валя. Задача Вали — найти и восстановить изменённое Машей число. Как могут действовать Маша и Валя, чтобы это произошло — несмотря на повороты массива по циклу?

В каждом тесте количество поворотов массива по циклу зафиксировано заранее, но держится в секрете.

Первый запуск

При первом запуске решение действует за Машу. В первой строке записано имя «Masha». Во второй строке задан массив: числа a_1, a_2, a_3, a_4, a_5 ($1 \leq a_i \leq 5$).

В ответ выведите массив a , в котором ровно одно число отличается на единицу от данного.

Второй запуск

При втором запуске решение действует за Валю. В первой строке записано имя «Valya». Во второй строке задан массив: числа b_1, b_2, b_3, b_4, b_5 . Это массив, выведенный при первом запуске, а после этого повернутый по циклу неизвестное количество раз.

В ответ выведите массив b , в котором ровно одно число отличается на единицу от данного: то число, которое поменяла Маша, должно стать исходным числом. Поворачивать массив по циклу не нужно.

Примеры

В каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске.

Далее показаны два запуска какого-то решения на первом тесте.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
Masha 1 1 1 1 1	1 1 0 1 1
Valya 0 1 1 1 1	1 1 1 1 1

Далее показаны два запуска какого-то решения на втором тесте.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
Masha 5 1 4 5 4	5 1 5 5 4
Valya 4 5 1 5 5	4 5 1 4 5

Система оценки

В задаче две группы тестов. Чтобы получить баллы за группу, нужно решить задачу на всех тестах, удовлетворяющих ограничениям этой группы.

В первой группе (40 баллов) все числа во вводе одинаковые.

Во второй группе (60 баллов) нет дополнительных ограничений.

Задача В. Нарисованный круг

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Это задача с открытыми тестами.

Витя решает следующую задачу:

Рассмотрим квадратную картинку из $n \times n$ клеток-пикселей. Каждый пиксел может быть либо белым, либо чёрным. Изначально все пиксели белые.

Зафиксируем три вещественных числа x , y и r — координаты центра круга и его радиус. Нарисуйте этот круг на картинке: каждый пиксел, центр которого лежит внутри круга, нужно покрасить в чёрный цвет.

Формально: противоположные углы картинки имеют координаты $(0, 0)$ и (n, n) . Центр пиксела в i -м слева столбце ($1 \leq i \leq n$) и j -й сверху строке ($1 \leq j \leq n$) имеет координаты $x' = i - 0.5$ и $y' = j - 0.5$. Центр находится внутри круга, если $(x' - x)^2 + (y' - y)^2 < r^2$.

Числа x , y и r выбираются так. Сначала случайно и равномерно выбирается центр круга. Затем радиус выбирается случайно и равномерно, но так, чтобы круг целиком лежал в квадрате. После этого, если картинка содержит меньше 10 чёрных пикселей, весь выбор начинается заново.

Решите обратную задачу: по кругу, который получился у Вити, найдите любую возможную тройку (x, y, r) , из которой он мог получиться.

Формат входных данных

В первой строке записано целое число n (во всех тестах $n = 20$).

Каждая из следующих n строк содержит n символов «.» и «#» — изображение круга, получившееся у Вити. Точка соответствует белым пикселям, решётка — чёрным.

Формат выходных данных

Выведите любые три вещественных числа x , y и r , для которых получается точно такой же круг. Круг как геометрическая фигура должен целиком лежать в квадрате $n \times n$.

Выводите числа с достаточным количеством знаков! Проверяющая программа использует 64-битный вещественный тип данных для всех вычислений (например, он называется `double` в C++ и `float` в Python).

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
20	12.37589 6.810178 4.029062
.....	
.....	
.....	
.....#####	
.....#####	
.....#####	
.....#####	
.....#####	
.....#####	
.....#####	
.....###	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	

Система оценки

Тесты оцениваются по отдельности. Однако, чтобы решение проверялось на основных тестах, нужно пройти пример.

В задаче один пример и 10 тестов. Каждый тест даёт по 10 баллов.

Тесты можно скачать по ссылкам:

- <https://tsweb.ru/olymp/2024-2025/raster-disc.tar.gz>
(переводы строк для Linux)
- <https://tsweb.ru/olymp/2024-2025/raster-disc.zip>
(переводы строк для Windows)

Задача С. Фабрики

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Вторник, первое января 1901 года. У Вилли есть одна фабрика и 0 монет. А ещё у Вилли есть хрустальный шар, предсказывающий стоимость постройки фабрик в следующие n лет: в год $1900 + i$ постройка каждой фабрики отнимает a_i монет.

В начале каждого года (весной) каждая фабрика Вилли приносит ему одну монету.

В конце каждого года (осенью) Вилли может построить любое количество новых фабрик, на которое ему хватает монет. Продавать фабрики нельзя.

Какое максимальное количество монет может быть у Вилли через n лет, в конце года $1900 + n$?

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100$).

Во второй строке заданы целые числа a_1, \dots, a_n — стоимость одной фабрики в каждый год ($2 \leq a_i \leq 10$).

Формат выходных данных

Выведите одно целое число: максимальное количество монет, которое может быть у Вилли через n лет.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 2 2 2 2 2	6
8 4 4 4 4 2 3 3 2	10

Система оценки

В задаче две группы тестов. Чтобы получить баллы за группу, нужно решить задачу на всех тестах, удовлетворяющих ограничениям этой группы.

В первой группе (30 баллов) все a_i равны.

Во второй группе (70 баллов) нет дополнительных ограничений.

Задача D. Сумма без соседей

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим массив a , изначально состоящий из n целых чисел. Можно выбрать в массиве любое подмножество позиций, среди которых нет двух соседних, и сложить элементы на этих позициях. Пусть $f(a)$ — максимальная сумма, которая может так получиться.

С массивом происходит q изменений. Каждое изменение имеет один из трёх типов:

- "insert v at p ": вставить значение v в позицию после p первых элементов массива;
- "assign v at p ": заменить на v значение в позиции после p первых элементов массива;
- "delete at p ": удалить значение в позиции после p первых элементов массива.

Найдите $f(a)$ для исходного массива, а также после каждого изменения.

Формат входных данных

В первой строке заданы два целых числа n и q ($0 \leq n, q \leq 200\,000$).

Во второй строке заданы целые числа a_1, \dots, a_n — исходный массив ($0 \leq a_i \leq 10^9$).

В каждой из следующих q строк задано очередное изменение в указанном выше формате (числа v и p целые; $0 \leq v \leq 10^9$; $0 \leq p$). Все изменения корректны.

Формат выходных данных

Выведите $q + 1$ целое число: значения $f(a)$ после $0, 1, 2, \dots, q$ первых изменений.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>массив</i>
8 6	19	<u>4</u> <u>1</u> <u>1</u> <u>2</u> <u>6</u> <u>0</u> <u>3</u> <u>8</u>
4 1 1 2 6 0 3 8	19	2 <u>4</u> <u>1</u> <u>1</u> <u>2</u> <u>6</u> <u>0</u> <u>3</u> <u>8</u>
insert 2 at 0	24	<u>9</u> <u>4</u> <u>1</u> <u>1</u> <u>2</u> <u>6</u> <u>0</u> <u>3</u> <u>8</u>
assign 9 at 0	24	<u>9</u> <u>4</u> <u>1</u> <u>1</u> <u>2</u> <u>6</u> <u>1</u> <u>3</u> <u>8</u>
assign 1 at 6	25	<u>9</u> <u>4</u> <u>1</u> <u>1</u> <u>2</u> <u>6</u> <u>5</u> <u>3</u> <u>8</u>
assign 5 at 6	19	<u>9</u> <u>4</u> <u>1</u> <u>1</u> <u>2</u> <u>6</u> <u>5</u> <u>3</u>
delete at 8	19	<u>9</u> <u>4</u> <u>1</u> <u>1</u> <u>6</u> <u>5</u> <u>3</u>
delete at 4		

Система оценки

В задаче три группы тестов. Чтобы получить баллы за группу, нужно решить задачу на всех тестах, удовлетворяющих ограничениям этой группы.

В первой группе (10 баллов) $q = 0$. Можно получить баллы за эту группу, не решив пример.

Во второй группе (30 баллов) $0 \leq n, q \leq 1000$.

В третьей группе (60 баллов) нет дополнительных ограничений.

Задача Е. Замощения доминошками и уголками

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Сколько способов замостить прямоугольник из $w \times h$ клеток доминошками и уголками?

Поскольку ответ может быть очень большим, найдите остаток от деления ответа на простое число 998 244 353.

Замечание

Доминошка — фигура из двух клеток, имеющих общую сторону.

Уголок — фигура из трёх клеток, в которой у каждой двух клеток есть либо общая сторона, либо общая вершина.

Замощение прямоугольника — набор фигур, для которого каждая клетка каждой фигуры лежит в прямоугольнике, а каждая клетка прямоугольника лежит ровно в одной фигуре.

Два замощения считаются различными, если какие-то две клетки прямоугольника в одном замощении лежат в одной и той же фигуре, а в другом замощении — в разных фигурах.

Формат входных данных

В первой строке заданы два целых числа w и h ($2 \leq w \leq 5$; $2 \leq h \leq 10^9$).

Формат выходных данных

Выведите одно целое число: остаток от деления количества замощений на 998 244 353.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3	5

Система оценки

В задаче четыре группы тестов. Чтобы получить баллы за группу, нужно решить задачу на всех тестах, удовлетворяющих ограничениям этой группы.

В первой группе (10 баллов) $h \leq 5$.

Во второй группе (20 баллов) $w = 2$ и $h \leq 1000$.

В третьей группе (30 баллов) $h \leq 1000$.

В четвёртой группе (40 баллов) нет дополнительных ограничений.

Задача F. Клика в плотном графе

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Это задача с оценкой решения по качеству.

Задан неориентированный граф из n вершин, пронумерованных целыми числами от 1 до n . Граф сгенерирован случайно: ребро между каждой парой различных вершин присутствует в графе с вероятностью 0.9, независимо от других рёбер.

Найдите в этом графе клику как можно большего размера.

Кликкой называется множество вершин, в котором любые две различные вершины соединены ребром.

Формат входных данных

В первой строке задано целое число n — количество вершин в графе. В примере $n = 10$, а во всех основных тестах $n = 1000$.

Каждая из следующих n строк содержит n двоичных цифр. Вместе эти строки задают матрицу смежности графа: элемент в i -й строке и j -м столбце равен 1, если в графе есть ребро между вершинами i и j , и равен 0 иначе.

Формат выходных данных

В первой строке выведите целое число k — количество вершин в клике ($1 \leq k \leq n$).

В следующей строке перечислите вершины клики по одному разу в любом порядке.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
10 0110111111 1011011111 1101111110 0110111111 1011010111 1111101011 1111010111 1111101011 1111111101 1101111110	6 1 2 3 8 9 7

Система оценки

В задаче один пример, в котором $n = 10$, и десять основных тестов, в которых $n = 1000$. В примере принимается любой корректный ответ.

Каждый ответ оценивается качеством: размером клики k . За каждый основной тест можно получить 10 баллов, если $k \geq 60$, и $10 \cdot (k/60)^3$ баллов иначе. Баллы за каждый тест округляются вниз с точностью до 0.05, а суммарные баллы — вниз до целого.