

Задача А. Ферзи

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На доске $n \times n$ расставьте, пожалуйста, n обычных шахматных ферзей так, чтобы они друг друга не били.

Формат входных данных

В единственной строке входных данных содержится одно целое число n — размер доски ($4 \leq n \leq 200$).

Формат выходных данных

Для каждой горизонтали исходной доски выведите номер вертикали, на которой стоит ферзь в этой горизонтали, вертикали нумеруются слева направо, начиная с единицы.

Пример

стандартный ввод	стандартный вывод
4	3 1 4 2

Задача В. Хорошие раскраски

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Назовем раскраску клеток таблицы $n \times m$ хорошей, если никакие четыре клетки, центры которых образуют вершины прямоугольника со сторонами, параллельными осям координат, не покрашены в один цвет. Иначе говоря, для раскраски не должно быть четверки целых чисел x_1, x_2, y_1, y_2 , что $1 \leq x_1 < x_2 \leq n, 1 \leq y_1 < y_2 \leq m$, и клетки $(x_1, y_1), (x_2, y_1), (x_1, y_2)$ и (x_2, y_2) покрашены в одинаковый цвет.

Требуется написать программу, которая по заданным целым числам n, m и c находит любую хорошую раскраску таблицы $n \times m$ в c цветов.

Формат входных данных

В первой строке записаны три целых числа n, m, c ($2 \leq n, m \leq 10, 2 \leq c \leq 3$).

Гарантируется, что для заданных во входных данных значений существует хотя бы одна хорошая раскраска.

Формат выходных данных

Выведите n строк по m чисел в каждой.

В качестве j -го числа i -й строки выведите $a_{i,j}$ — цвет клетки (i, j) ($1 \leq a_{i,j} \leq c$).

Если есть несколько хороших раскрасок, можно вывести любую из них.

Пример

стандартный ввод	стандартный вывод
2 2 2	1 2 2 2

Замечание

Примените какой-нибудь неточный алгоритм нахождения минимума функции, например алгоритм отжига.

Задача D. SO-SAT

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Найдите решение 3-SAT. Гарантируется, что оно 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 ($1 \leq n \leq 90, 1 \leq m \leq \min(n^2, 1000)$).

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.

Пример

стандартный ввод	стандартный вывод
2 3 1 0 1 0 1 0 2 0 2 1 1 1 1 1 2 1 1 1	01

Задача Е. Проще всего красить в три цвета

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — покрасить его вершины в три цвета таким образом, чтобы смежные вершины были покрашены в разные цвета. Гарантируется, что покрасить граф в три цвета возможно.

Формат входных данных

На первой строке число вершин n ($1 \leq n \leq 250$) и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходных данных

На следующей строке n целых чисел от 1 до 3 — цвета вершин. Если требуемых раскрасок несколько, выведите любую.

Пример

стандартный ввод	стандартный вывод
5 5 1 2 2 3 3 1 4 5 1 5	1 2 3 1 2

Задача F. Как разрушить свой мозг

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Победителей не судят. Их приглашают в жюри

девиз Воронежской олимпиады

Пока Сквив, Гвидо и Нунцио регистрировались в качестве участников, к Аазу подошёл представитель организаторов и спросил:

- Вы приехали как тренер?
- Да, а разве Вам не пришла заявка? Команда Possiltum U M.Y.T.H. . . .
- Конечно пришла. Просто у нас тут возникла идея. Раз Вы, во-первых, тренер, а во-вторых, как я вижу, извращенец, то мы приглашаем Вас принять участие в работе жюри.
- Значит так. Во-первых, я не извращенец, а изверг. . . — сурово уточнил Ааз.
- Тем более, у нас в жюри как раз не хватает сторонников строгого подхода. Так что давайте, жюри Вас ждёт.

Ааз решил, что лишняя информация о ходе соревнований ему не повредит, и принял приглашение, предварительно осведомившись о вознаграждении, положенном членам жюри за работу.

Вчитываясь в условие очередной задачи, как раз и посвящённой вычитыванию условий задач, Ааз почувствовал, что этот текст начинает разрушать его мозг. Он вспомнил, что изверг может разрушить свой мозг чтением строки S , если хотя бы половина из всех её k -буквенных подпоследовательностей совпадает.

Чтобы уберечь мозг ценного сотрудника корпорации «МИФ», напишите программу, которая по заданной строке S выясняет, может ли Ааз разрушить свой мозг, читая эту строку.

Формат входных данных

В первой строке дана непустая строка S , длиной до 1 000 маленьких букв латинского алфавита. Во второй — натуральное число $1 \leq k \leq |S|$.

Формат выходных данных

Выведите «YES», если строка может разрушить мозг изверга, и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
aaaaaaa 3	YES
abc 2	NO

Задача G. Красивый отчёт

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 1024 мегабайта

Андрей занимается анализом графа подписок в одной социальной сети. Этот граф является ориентированным: если пользователь a подписан на пользователя b , то пользователь b не обязательно подписан на пользователя a .

Менеджер Андрея попросил его посчитать для каждого пользователя x , сколько существует пользователей y , таких, что от пользователя x можно добраться в графе подписок до пользователя y .

Печатать точное значение не имеет смысла, потому что оно смотрится некрасиво и моментально устареет, поэтому, вас интересует лишь примерное значение. Выполните задание менеджера и найдите эти значения с ошибкой не более чем в два раза.

Формат входных данных

В первой строке заданы целые числа n и m ($1 \leq n, m \leq 200\,000$) — число пользователей социальной сети и число ситуаций, когда один пользователь подписан на другого.

Далее, в m строках идёт описание графа, i -я из этих строк содержит два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) и означает, что пользователь a_i подписан на пользователя b_i . Каждая упорядоченная пара (a_i, b_i) встречается во входных данных не более одного раза.

Формат выходных данных

Выведите n целых чисел q_i — оценку на количество пользователей y , таких, что от пользователя i можно добраться в графе подписок до пользователя y . Если настоящее количество таких пар равно z_i , должны выполняться неравенства $q_i \leq 2z_i$ и $z_i \leq 2q_i$. Кроме того, допустимо для не более, чем 10 пользователей вывести q_i , не удовлетворяющее указанным ограничениям.

Примечание: мы используем для проверки ваших ответов точное количество искомым пользователей. При этом мы не гарантируем, что точное число можно найти, уложившись в ограничения по времени и памяти, и рекомендуем использовать возможность вывести приблизительное значение.

Пример

стандартный ввод	стандартный вывод
5 6	7
1 2	3
1 3	2
2 4	2
2 5	1
3 5	
4 2	

Замечание

В примере от пользователя 1 можно добраться до всех пяти пользователей. Однако, показанный ответ 7 тоже допустим, так как отличается от 5 не более, чем в два раза. Аналогично, допустимым является ответ 2 для пользователя 4.

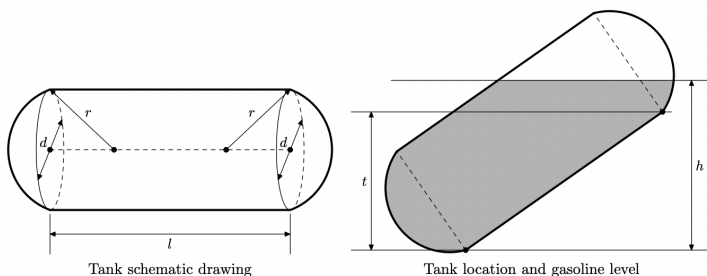
Задача Н. Заполняем Таблетку

Имя входного файла: `damage.in`
Имя выходного файла: `damage.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Вам дана фигура цилиндра с двумя полусферическими насадками на плоских сторонах, заполненная водой до какого-то уровня.

Диаметр цилиндра равен d , высота равна l . Полусферы имеют радиус r , $2r \geq d$. Вместо угла наклона фигуры вам дана разница высот между двумя нижними точками верхней и нижней плоской части цилиндра.

В цилиндр налили жидкости и замерили расстояние h от нижней точки цилиндра до уровня жидкости. Гарантируется, что уровень жидкости пересекает цилиндр ($0 \leq h \leq t + d\sqrt{1 - (t/l)^2}$).



Найдите объем жидкости в цилиндре.

Формат входных данных

На ввод подаются пять целых чисел — d , l , r , t и h , задающие диаметры длину цилиндра, радиус полусфер, наклон и уровень жидкости соответственно. Все в миллиметрах (10^{-3} метра). Также $d, l \geq 100$, $d, l, r \leq 10\,000$

Формат выходных данных

Выведите одно вещественное число — объем жидкости в литрах (1 литр = 10^{-3} кубических метров).

Абсолютная ошибка не должна превышать 0.1 литра.

Примеры

<code>damage.in</code>	<code>damage.out</code>
3000 6000 1600 0 3000	50974.555972040616
3000 6000 1600 3441 4228	40728.89794364132

Замечание

Обратите внимание, что ввод и вывод в этой задаче осуществляется с помощью файлов.

Задача I. Фибоначчиева куча

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Эта задача для тех, кто завалил регион (или его дисквалифицировали), отчаялся в олимпиадной проге и решил полностью посвятить себя теоретической информатике и продвинутым алгоритмам. Вам предстоит написать самую настоящую Фибоначчиеву кучу. Все запросы будут задаваться массивом a_i , где a_1 вам дано изначально, а для $i > 1$ $a_i = (a_{i-1} * b + c) \bmod 2^{32}$.

К вам будут поступать запросы двух типов:

1. В случае, если a_i чётно, то в i -м запросе от вас будет требоваться добавить число a_i в кучу.
2. В случае, если a_i нечётно от вас будет требоваться узнать значение верхнего (максимального) элемента в куче.

Формат входных данных

В первой строке вам дано единственное число n ($1 \leq n \leq 10^8$). В следующей строке вам дано 3 числа a_1 b и c ($0 \leq a_1, b, c \leq 2^{32}$). Гарантируется, что a_1 чётно.

Формат выходных данных

Для всех запросов второго типа выведите сумму ответов на них.

Пример

стандартный ввод	стандартный вывод
10 2 2 1	18

Замечание

Так как Фибоначчиева куча сложная и её код трудно читать, в этой задаче не будет ревью кода.

Задача J. Биномиальная куча

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте биномиальную кучу.

Формат входных данных

В первой строке содержится два целых числа: N — общее количество куч и M — количество операций ($1 \leq N \leq 1000, 1 \leq M \leq 1\,000\,000$). Изначально все кучи пусты.

Требуется поддерживать следующие операции:

- 0 a v — добавить элемент со значением v в кучу с номером a . Вновь добавленный элемент имеет уникальный индекс равный порядковому номеру соответствующей операции добавления. Нумерация начинается с единицы.
- 1 a b — переложить все элементы из кучи с номером a в кучу с номером b . После этой операции куча a становится пустой.
- 2 i — удалить элемент с индексом i .
- 3 i v — присвоить элементу с индексом i значение v . Гарантируется, что элемент существует.
- 4 a — вывести на отдельной строке значение минимального элемента в куче с номером a . Гарантируется, что куча не пуста.
- 5 a — удалить минимальный элемент из кучи с номером a . Если таковых несколько, то выбирается элемент с минимальным индексом. Гарантируется, что куча не пуста.

Формат выходных данных

Для каждой операции поиска минимального элемента выведите единственное число: значение искомого элемента.

Пример

стандартный ввод	стандартный вывод
3 19	10
0 1 10	5
4 1	7
0 2 5	7
0 2 7	10
4 2	3
3 2 20	10
4 2	8
1 2 1	
4 1	
5 1	
4 1	
3 2 3	
4 1	
2 2	
4 1	
0 1 9	
1 1 3	
0 3 8	
4 3	

Задача К. Персистентная приоритетная очередь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Требуется реализовать структуру данных, которая хранит мультимножество и умеет изменять любую свою предыдущую версию, выполняя одну из этих операций:

1. Заданы v и x , требуется добавить в множество v элемент со значением x , после чего вывести минимальный элемент в получившемся множестве.
2. Заданы v и u , требуется объединить множества с номерами v и u , после чего вывести минимальный элемент в получившемся множестве.
3. Задано v , требуется вывести минимальный элемент в множестве v , после чего удалить минимальный элемент из множества v . Если множество пустое, то вывести, что множество пустое, и создать новое пустое множество.

Изначально есть одно пустое множество с номером 0. После операции с номером i множество, получаемое во время этой операции, получает номер i .

Формат входных данных

Первая строка содержит число n — количество операций для выполнения.

От вас потребуется отвечать на запросы в онлайн, при этом поддерживая переменную s . Она изначально равна нулю. После каждой операции, она пересчитывается следующим образом через предыдущее значение: если ответ на запрос равен x , то $s = (s_{old} + x) \bmod 239017$. Если же ответом на запрос является слово `empty`, то s не изменяется.

В следующих n строках заданы запросы.

Запросы первого типа описываются строкой `1 a b`, где a и b — неотрицательные целые числа, которые описывают v и x для соответствующего запроса, как $v = (a + s) \bmod i$ и $x = (b + 17s) \bmod (10^9 + 1)$, где i — номер соответствующего запроса.

Запросы второго типа описываются строкой `2 a b`, где a и b — неотрицательные целые числа, которые описывают v и u для соответствующего запроса, как $v = (a + s) \bmod i$ и $u = (b + 13s) \bmod i$, где i — номер соответствующего запроса.

Запросы третьего типа описываются строкой `3 a`, где a — неотрицательное целое число, которые описывает v для соответствующего запроса, как $v = (a + s) \bmod i$, где i — номер соответствующего запроса.

Число запросов не превышает 200 000. Гарантируется, что мощность любого созданного мультимножества не превышает 2^{63} .

Формат выходных данных

Требуется вывести ровно n строк, в каждой строке должно находиться неотрицательное целое число либо слово `empty`.

Для запросов первого и второго типа требуется вывести значение минимального элемента в только что созданном множестве, либо слово `empty`, если множество пустое.

Для запросов третьего типа требуется вывести минимальный элемент в множестве, либо слово `empty`, если множество пустое.

Пример

стандартный ввод	стандартный вывод
9	2
1 0 2	3
1 0 999999970	2
2 2 0	2
3 0	2
2 4 4	2
3 0	2
3 0	3
3 0	empty
3 8	