



Общая информация по задачам олимпиады

Ограничение по памяти

Во всех задачах ограничение составляет 512 МБ.

Ограничение на размер исходного кода программы

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

Ограничение на посылку решений

По каждой задаче на проверку принимается не более 50 решений.

По каждой задаче участник не может отправить решение более одного раза в течение 30 секунд. Это ограничение не распространяется на последние 15 минут соревнований.

Система оценки

Каждая задача олимпиады поделена на несколько подзадач. Чтобы набрать баллы по подзадаче, программа должна пройти все тесты этой подзадачи.

За каждую задачу выставляется суммарный балл по всем ее подзадачам. В каждой подзадаче оценивается лучшее решение, то есть за подзадачу выставляется максимальный набранный по ней балл среди всех решений.

Получение информации о результатах проверки

Чтобы получить информацию о проверке вашего решения, используйте ссылку «Информация о проверке» во вкладке «Решения» в PCMS2 Web Client. По каждой задаче вам будет доступна информация по количеству набранных баллов в каждой подзадаче или результат проверки на первом непройденном тесте.

Таблица результатов

Во время соревнования доступна текущая таблица результатов. Для доступа к ней используйте ссылку «Результаты» в PCMS2 Web Client. Таблица результатов в PCMS2 Web Client не является окончательной.

Интерактивные задачи

После каждого действия вашей программы выводите перевод строки.

После каждого действия вашей программы делайте сброс потока вывода.

Если вы используете `writeln` в Паскале, `cout << ... << endl` в C++, `System.out.println` в Java, `print` в Python, `Console.WriteLine` в C#, то сброс потока вывода у вас происходит автоматически, дополнительно ничего делать не требуется. Если вы используете другой способ вывода, рекомендуется делать сброс потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае. Для сброса потока вывода можно использовать `fflush(stdout)` в C и C++, `flush(output)` в Паскале, `System.out.flush()` в Java, `sys.stdout.flush()` в Python, `Console.Out.Flush()` в C#. Обратите внимание, что в Borland Delphi `flush` делать обязательно.

Типичные ошибки в интерактивных задачах:

- «Wrong Answer» означает, что ответ или промежуточные действия неверны, либо что ваша программа нарушила протокол. Обратите внимание, что результата проверки
- «Idleness Limit Exceeded» означает, что ваша программа ожидает ввода, но данных в стандартном потоке ввода нет. Например,
 - ваша программа ошибочно ожидает ввода, а она должна вывести информацию для программы жюри либо завершиться;
 - ваша программа не вывела перевод строки или не произвела сброс потока вывода, программа жюри не получила вывод вашей программы и не может выполнить свои действия.
- «Runtime Error» редко означает проблемы с интерактивностью и чаще возникает из-за обычных ошибок в программе.



Задача А. Designing a New Logo

Ограничение по времени: 2 секунды

«Тинькофф» проводит ребрендинг! В силу последних трендов логотипов компаний, было решено, что новый логотип будет в *шахматной раскраске*. Это значит, что логотип будет являться фигурой, состоящей из квадратов 1×1 , причем любые два квадрата, соседних по стороне, имеют разные цвета, один из них черный, а другой — белый. Логотип также должен быть *связным*: от любого квадрата до любого другого можно прийти, переходя на соседний по стороне квадрат.

Брендбук предписывает, что в логотипе должно быть w белых клеток и b черных. Чтобы логотип не выглядел слишком темным (а также для упрощения вашей работы), верно, что $b \leq w \leq 3b + 1$.

Для того, чтобы создать новые логотипы «Тинькофф», были закуплены листы в шахматной раскраске, каждый размером $4n$ строк на $4m$ столбцов, из которых вы будете вырезать логотипы. Левая верхняя клетка каждого листа раскрашена в черный цвет. На всякий случай, размеры листов выбрали с запасом, чтобы логотип наверняка влез: число черных клеток b не превосходит nm .

Ваша задача в том, чтобы разработать связный логотип из b черных и w белых клеток так, чтобы он помещался на заготовку размера $4n \times 4m$. Аналитики компании доказали вам, что в таких ограничениях всегда возможно сделать логотип, подходящий под условия $b \leq w \leq 3b + 1$ и $b \leq nm$.

Формат входных данных

Входные данные содержат несколько тестов.

В первой строке задано целое число t — число тестов во входных данных ($1 \leq t \leq 2000$). Далее идут t тестов, каждый в отдельной строке.

Каждая строка содержит четыре целых числа n, m, b, w ($1 \leq n, m \leq 50$; $0 \leq b \leq n \cdot m$; $b \leq w \leq 3b + 1$).

Сумма $4n \cdot 4m$ по всем тестам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого теста выведите матрицу размером $4n \times 4m$, на которой изображен черно-белый логотип. Символом «W» обозначьте белые клетки фигуры, символом «B» — чёрные клетки фигуры, а символом «.» — клетки листа, не принадлежащие логотипу.

Если существует несколько возможных решений, выведите любое.

Система оценки

Подзадача	Баллы	Ограничения
1	20	$b = w$
2	25	$b \leq 2m - 1$
3	25	$b \leq 2(n + m) - 3$
4	30	Без дополнительных ограничений



Пример

стандартный ввод	стандартный вывод
2
2 3 5 7	..WBWBWB...
2 2 4 8W.....
B.....
W.....
B.....
W.....

	...W.W..
	..WBWB..
	...W.W..
	..WBWB..



Задача В. Even Tree

Ограничение по времени: 2 секунды

Дан связный неориентированный взвешенный граф с n вершинами и m ребрами. Выберите в нем ровно $n - 1$ ребро так, чтобы граф все еще оставался связным и суммарный вес выбранных ребер был четным, либо скажите, что это невозможно.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 3 \cdot 10^5$; $n - 1 \leq m$) — количество вершин и ребер, соответственно.

Следующие m строк содержат информацию о ребрах. $(i + 1)$ -я строка входных данных содержит три целых числа v_i, u_i, w_i ($1 \leq v_i, u_i \leq n$; $0 \leq w_i \leq 10^9$) — начало и конец i -го ребра и его вес, соответственно. В графе могут быть кратные ребра и петли.

Формат выходных данных

В случае, если выбрать ребра так нельзя, выведите строку «NO».

Иначе выведите строку «YES» и номера ребер, которые следует выбрать.

Если правильных ответов несколько, выведите любой из них.

Система оценки

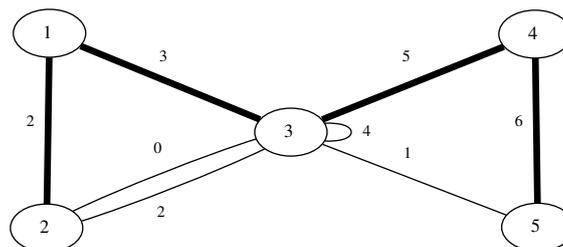
Подзадача	Баллы	Ограничения
1	18	$n, m \leq 10$
2	23	все w_i четные
3	16	$n = m$
4	19	$n, m \leq 1000$
5	24	Без дополнительных ограничений

Пример

стандартный ввод	стандартный вывод
5 8 1 2 2 2 3 0 2 3 2 3 1 3 3 4 5 4 5 6 5 3 1 3 3 4	YES 1 4 5 6

Замечание

В первом тесте из примера мы выбрали ребра (1, 2), (3, 1), (3, 4) и (4, 5). Сам граф выглядит так:





Задача C. Primle

Ограничение по времени: 3 секунды

Это интерактивная задача.

Жюри загадало пятизначное простое число (без ведущих нулей). Ваша задача состоит в том, чтобы отгадать это число. Для вашей каждой догадки программа жюри вернет вам информацию о том, какие цифры уже стоят на своем месте, а какие цифры есть в этом числе, но находятся в другом разряде.

Например, если жюри загадало число 19937, а ваша догадка — 31357, то ответом жюри будет ??--+: последняя цифра 7 стоит на своем месте, а первые две цифры присутствуют в числе, но не стоят на своих местах. Обратите внимание, что второе вхождение цифры 3 помечено, как -, потому что в загаданном числе 3 встречается только один раз.

Более формально: пусть жюри загадало число $S = s_1s_2s_3s_4s_5$, а ваша догадка это $T = t_1t_2t_3t_4t_5$. Во всех позициях i , где $s_i = t_i$, вам вернется +. Все такие позиции будем считать использованными. Для оставшихся позиций, слева направо, выполняется следующая проверка: если цифра t_i присутствует среди еще не использованных цифр в S , то в i -й позиции ответа будет стоять ?, а соответствующая цифра в S помечается, как использованная. Во всех остальных позициях в ответе жюри будет стоять -.

В этой игре есть два режима: простой и сложный. В простом режиме вашей догадкой может быть любая строка из пяти цифр (ведущие нули разрешены). В сложном режиме вы можете спрашивать только про пятизначные простые числа без ведущих нулей. Для каждой подзадачи есть определенный лимит запросов, в который нужно уложиться.

В каждом тесте будет T раундов, в каждом из которых вам нужно будет отгадать число. Набор загаданных чисел для каждого теста фиксирован до начала олимпиады.

Протокол взаимодействия

В самом начале взаимодействия вы должны считать строку, сообщающую, в каком режиме проходит игра — простом или сложном, а также максимальное число догадок L , которые вы можете использовать для каждого числа. Простой режим обозначается строкой «easy», а сложный режим — строкой «hard».

Затем, в следующей строке вы должны считать число T ($1 \leq T \leq 10$) — количество раундов, в которых вам предстоит отгадать простое число.

Каждый раунд состоит из того, что вы отправляете строку из пяти цифр. В простом режиме на эту строку нет ограничений, а в сложном режиме эта строка должна задавать пятизначное простое число без ведущих нулей. В ответ от программы жюри вы получите строку из пяти символов -, ?, или +. Если же вы нарушили формат взаимодействия, либо на L -м запросе раунда вы не угадали число, вы получите в ответ строку !!!!!. После получения этого ответа вам необходимо завершить исполнение программы.

После того, как вы получили ответ +++++, начинается следующий раунд. После окончания последнего раунда вам необходимо завершить исполнение программы. Не забывайте выводить перевод строки после каждой догадки.

Система оценки

Подзадача	Баллы	Сложный режим?	Ограничения	Необх. подзадачи
1	7	–	$L = 100$	–
2	11	+	$L = 100$	1
3	8	–	$L = 10$	1
4	15	+	$L = 10$	1,2,3
5	12	–	$L = 7$	1,3
6	16	+	$L = 7$	1,2,3,4,5
7	14	–	$L = 5$	1,3,5
8	17	+	$L = 5$	1,2,3,4,5,6,7



Примеры

стандартный ввод	стандартный вывод
easy 10 1 -??+- -+---? +++++	12345 78778 28843
hard 5 1 ??--+ ?+--? +++++	31357 99083 19937



Задача D. Add and Multiply

Ограничение по времени: 2 секунды

Даны два массива целых чисел a и b размера n . Найдите такой массив целых неотрицательных чисел c длины n , что если прибавить его поэлементно к a и b , то произведения чисел в новых массивах будут равны, либо скажите, что это невозможно.

Более формально, вам нужно найти целые неотрицательные числа c_1, c_2, \dots, c_n , такие что $(a_1 + c_1) \cdot (a_2 + c_2) \cdot \dots \cdot (a_n + c_n) = (b_1 + c_1) \cdot (b_2 + c_2) \cdot \dots \cdot (b_n + c_n)$.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 100\,000$) — количество чисел в массивах a и b .

Во второй строке находятся n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1000$).

В третьей строке находятся n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 1000$).

Формат выходных данных

Если ответа не существует, выведите «NO». Иначе, в первой строке выведите «YES», а в следующей строке — n целых чисел c_1, c_2, \dots, c_n ($0 \leq c_i \leq 10^{18}$) — решение задачи. Если решений несколько — выведите любое. Гарантируется, что если решение существует, то существует и решение, подходящее под ограничения.

Система оценки

Подзадача	Баллы	Ограничения
1	9	$n \leq 2; a_i, b_i \leq 10$
2	25	$n \leq 2$
3	19	$n \leq 3$
4	18	$a_i, b_i \leq 2$
5	29	Без дополнительных ограничений

Пример

стандартный ввод	стандартный вывод
4 1 2 5 4 6 1 6 2	YES 6 4 1 1



Задача E. Draft Laws

Ограничение по времени: 2 секунды

Мечта Юлии наконец-то сбылась: она стала руководителем Отдела Общего Управления Иннополиса (ООУИ)! В её отделе всё устроено наиболее современным образом: документы разрабатываются на компьютерах с использованием программного обеспечения последнего поколения.

В Отделе имеется n рабочих компьютеров, к каждому из которых прикреплен уникальный номер от 1 до n и опытный сотрудник, работающий только на нём. Некоторые пары компьютеров соединены кабелем, позволяющим передавать данные между этими компьютерами в обе стороны. При этом, между любыми двумя компьютерами существует ровно один способ передать данные по кабелям, не используя никакой кабель дважды. Иными словами, сеть компьютеров и кабелей образует дерево.

В настоящий момент в отделе разрабатывается k законопроектов. Юлии нужно составить план на завтра: она должна для каждого компьютера определить, какой из k законопроектов будет править сотрудник за этим компьютером. Каждый сотрудник должен работать ровно над одним законопроектом в течение дня, но, при этом, один законопроект может одновременно редактироваться на нескольких компьютерах (или даже ни на одном). Некоторые сотрудники имеют свой любимый законопроект, и они согласны редактировать только его. Кроме того, в соответствии с законодательством, **нельзя допустить, чтобы на двух компьютерах, соединённых кабелем, редактировался один и тот же законопроект.**

Юля очень взволнована перед своим первым рабочим днем, и ей интересно, сколько у неё есть способов составить план на завтра, удовлетворив все пожелания сотрудников и требования законодательства. Помогите ей найти это число способов! Так как искомое число может быть большим, найдите его по модулю $10^9 + 7$.

Формат входных данных

Первая строка содержит два числа n и k ($1 \leq n \leq 3 \cdot 10^5$, $2 \leq k \leq 10^9$) — число компьютеров в Отделе Общего Управления Иннополиса и число разрабатываемых законопроектов соответственно.

Во второй строке находятся n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq k$), где a_i — номер любимого законопроекта сотрудника за компьютером i , или 0, если он согласен редактировать любой законопроект.

В каждой из следующих $n - 1$ строк дано по два целых числа u_j и v_j — номера компьютеров, соединённых кабелем ($1 \leq u_j, v_j \leq n$; $u_j \neq v_j$). Гарантируется, что сеть компьютеров и кабелей образует дерево.

Формат выходных данных

В единственной строке выведите одно целое число — количество способов составить план на завтра, удовлетворяющий всем требованиям, взятое по модулю $10^9 + 7$.

Система оценки

Подзадача	Баллы	Ограничения
1	5	$n, k \leq 8$
2	5	$k = 2$
3	5	все a_i равны 0
4	10	$n, k \leq 300$
5	15	$n, k \leq 3000$
6	20	$n \leq 3000$
7	10	Каждый компьютер соединён кабелем с не более, чем двумя другими компьютерами
8	20	$n \leq 10^5$
9	10	Без дополнительных ограничений



Примеры

стандартный ввод	стандартный вывод
4 3 0 0 1 2 1 2 2 3 2 4	2
4 3 1 0 0 0 1 2 2 3 2 4	8
4 1000000000 0 2 2 0 1 2 2 3 3 4	0

Замечание

В первом примере на компьютерах 3 и 4 уже определены законопроекты 1 и 2, поэтому для компьютера 2 есть только один возможный законопроект: 3. А для компьютера 1 можно выбрать два законопроекта (1 и 2), поэтому общее число способов равно 2.

В третьем примере на двух соседних компьютерах 2 и 3 уже определен один и тот же законопроект 2. Следовательно, никакой план заведомо не может удовлетворять всем ограничениям.