

Технокубок 2020 - Отборочный Раунд 2

А. Забывчивость

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Коля очень забывчивый. Сегодня учитель математики задал ему простой пример с равенством $a+1=b$ с положительными целыми числами a и b , но Коля забыл числа a и b . Однако, он помнит, что первая (самая левая) цифра a была d_a , а первая (самая левая) цифра b была d_b .

Можете ли вы восстановить любое равенство $a+1=b$ с этим свойством? Может быть такое, что Коля перепутал цифры, и решения не существует, в таком случае требуется это сообщить.

Входные данные

В единственной строке через пробел записаны две цифры d_a и d_b ($1 \leq d_a, d_b \leq 9$).

Выходные данные

Если не существует равенства $a+1=b$ с положительными целыми числами a и b , такое что первая цифра a равна d_a , а первая цифра b равна d_b , выведите одно число -1 .

В противном случае, выведите любые подходящие числа a и b , **каждое из которых** положительно и не превосходит 10^9 . Гарантируется, что

если решение существует, то существует и решение, в котором оба числа не превосходит 10^9 .

Пример

входные данные
1 2
выходные данные
199 200

входные данные
4 4
выходные данные
412 413

входные данные
5 7
выходные данные
-1

входные данные
6 2
выходные данные
-1

В1. Покупка подписок на сериалы (упрощённая версия)

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Единственное отличие между простой и сложной версиями — ограничения.

Канал BeTV каждый день показывает один эпизод одного из k сериалов. Вам известно расписание на ближайшие n дней: последовательность целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq k$), где a_i — номер сериала, эпизод которого будет показан в i -й день.

Подписка на сериалы покупается целиком полностью на весь сериал, на каждый сериал подписка покупается отдельно.

Сколько минимум подписок надо купить, чтобы была возможность d ($1 \leq d \leq n$) дней подряд смотреть эпизоды купленных по подписке сериалов? Иными словами, вы хотите купить минимальное количество сериалов, чтобы существовал некоторый отрезок из d подряд идущих дней, в котором все эпизоды из сериалов по приобретённым подпискам.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 100$) — количество наборов входных данных в тесте. Далее следуют описания t наборов входных данных.

Первая строка каждого из наборов содержит три целых числа n, k и d ($1 \leq n \leq 100$, $1 \leq k \leq 100$, $1 \leq d \leq n$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq k$), где a_i — это номер сериала, который показывается в i -й день.

Гарантируется, что сумма значений n по всем наборам входных данных в тесте не превосходит 100 .

Выходные данные

Выведите t целых чисел — ответы на заданные наборы входных данных в порядке их следования в тесте. Ответом на набор входных данных является минимальное количество сериалов, подписку на которые надо приобрести, чтобы была возможность d дней подряд смотреть по BeTV эпизодов приобретённого сериала. Обратите внимание, что допустимо, что у будет возможность посмотреть более d дней подряд.

Пример

входные данные
4
5 2 2
1 2 1 2 1
9 3 3
3 3 3 2 2 2 1 1 1
4 10 4
10 8 6 4
16 9 8
3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3
выходные данные
2
1
4
5

Примечание

В первом наборе входных данных примера чтобы была возможность смотреть два дня подряд сериалы, обязательно нужно купить подписку и на сериал 1, и на сериал 2. Таким образом, ответ равен двум.

Во втором наборе входных данных примера можно купить подписку на любой из трёх сериалов, так как для каждого сериала есть отрезок из трех подряд дней, состоящий только из эпизодов этого сериала.

В третьем наборе входных данных примера в единственном отрезке из четырех дней встречаются четыре разных сериала, поэтому нужно приобрести подписку на все эти четыре сериала.

В четвертом наборе входных данных примера можно купить подписки на сериалы 3, 5, 7, 8, 9, таким образом можно будет смотреть сериалы в течении последних восьми дней.

V2. Покупка подписок на сериалы (усложнённая версия)

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Единственное отличие между простой и сложной версиями — ограничения.

Канал BeTV каждый день показывает один эпизод одного из k сериалов. Вам известно расписание на ближайшие n дней: последовательность целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq k$), где a_i — номер сериала, эпизод которого будет показан в i -й день.

Подписка на сериалы покупается целиком полностью на весь сериал, на каждый сериал подписка покупается отдельно.

Сколько минимум подписок надо купить, чтобы была возможность d ($1 \leq d \leq n$) дней подряд смотреть эпизоды купленных по

подписке сериалов? Иными словами, вы хотите купить минимальное количество сериалов, чтобы существовал некоторый отрезок из d подряд идущих дней, в котором все эпизоды из сериалов по приобретённым подпискам.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 10000$) — количество наборов входных данных в тесте. Далее следуют описания t наборов входных данных.

Первая строка каждого из наборов содержит три целых числа n, k и d ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq 10^6$, $1 \leq d \leq n$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq k$), где a_i — это номер сериала, который показывается в i -й день.

Гарантируется, что сумма значений n по всем наборам входных данных в тесте не превосходит $2 \cdot 10^5$

Выходные данные

Выведите t целых чисел — ответы на заданные наборы входных данных в порядке их следования в тесте. Ответом на набор входных данных является минимальное количество сериалов, подписку на которые надо приобрести, чтобы была возможность d дней подряд смотреть по BeTV эпизодов приобретённого сериала. Обратите внимание, что допустимо, что у будет возможность посмотреть более d дней подряд.

Пример

входные данные
4
5 2 2
1 2 1 2 1
9 3 3
3 3 3 2 2 2 1 1 1

4 10 4
10 8 6 4
16 9 8
3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3

ВЫХОДНЫЕ ДАННЫЕ

2
1
4
5

Примечание

В первом наборе входных данных примера чтобы была возможность смотреть два дня подряд сериалы, обязательно нужно купить подписку и на сериал 1, и на сериал 2. Таким образом, ответ равен двум.

Во втором наборе входных данных примера можно купить подписку на любой из трёх сериалов, так как для каждого сериала есть отрезок из трех подряд дней, состоящий только из эпизодов этого сериала.

В третьем наборе входных данных примера в единственном отрезке из четырех дней встречаются четыре разных сериала, поэтому нужно приобрести подписку на все эти четыре сериала.

В четвертом наборе входных данных примера можно купить подписки на сериалы 3, 5, 7, 8, 9, таким образом можно будет смотреть сериалы в течении последних восьми дней.

С. p -двоичные числа

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Васе нравятся любые числа, лишь бы они были целыми степенями двойки. Петя, в отличие от Васи, более консервативен, и ему нравится только одно число p (оно может быть положительным, отрицательным или нулём). Они решили объединить свои вкусы и изобрели p -двоичные числа вида 2^{x+p} , где x — неотрицательное целое число.

Например, некоторые -9 -двоичные («минус девять» двоичные) числа — это -8 (минус восемь), 7 и 1015 ($-8=2^0-9$, $7=2^4-9$, $1015=2^{10}-9$).

Теперь ребята используют p -двоичные числа, чтобы представлять всё вокруг. Они столкнулись с проблемой: для данного положительного целого n , какое наименьшее количество p -двоичных чисел (не обязательно различных) им понадобится, чтобы представить n в виде их суммы? Возможно, что представления вообще не существует. Помогите им справиться с этой задачей.

Например, при $p=0$ число 7 можно представить как $2^0 + 2^1 + 2^2$.

А при $p=-9$ число 7 можно представить одним числом (2^4-9).

Обратите внимание, что отрицательные p -двоичные числа разрешается использовать в сумме (пример есть в секции Примечание).

Входные данные

В единственной строке записано два целых числа n и p ($1 \leq n \leq 10^9$, $-1000 \leq p \leq 1000$).

Выходные данные

Если n нельзя представить как сумму какого угодно количества p -двоичных чисел, выведите одно число -1 . В противном случае, выведите минимальное число слагаемых.

Пример

входные данные
24 0
выходные данные
2

входные данные
24 1
выходные данные
3

входные данные
24 -1
выходные данные
4

входные данные
4 -7
выходные данные
2

входные данные
1 1
выходные данные
-1

Примечание

0-двоичные числа — это обычные степени двойки, поэтому в первом примере мы можем представить $24=(2^4+0)+(2^3+0)$.

Во втором примере мы можем представить $24=(2^4+1)+(2^2+1)+(2^0+1)$.

В третьем примере мы можем представить $24=(2^4-1)+(2^2-1)+(2^2-1)+(2^2-1)$. Обратите внимание, что одинаковые слагаемые разрешены.

В четвёртом примере мы можем представить $4=(2^4-7)+(2^1-7)$. Обратите внимание, что второе слагаемое отрицательное, и это разрешено.

В пятом примере ни одного представления не существует.

D. Произведения-степени

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод
вывод: стандартный вывод

Вам дано n целых положительных чисел a_1, \dots, a_n и целое $k \geq 2$.

Посчитайте количество пар i, j , таких что $1 \leq i < j \leq n$, а также существует целое x , такое что $a_i \cdot a_j = x^k$.

Входные данные

В первой строке записано два целых числа n и k ($2 \leq n \leq 10^5$, $2 \leq k \leq 100$).

Во второй строке записано n целых чисел a_1, \dots, a_n ($1 \leq a_i \leq 10^5$).

Выходные данные

Выведите одно число — количество подходящих пар.

Пример

входные данные
6 3
1 3 9 8 24 1
выходные данные
5

Примечание

В примере подходят следующие пары:

- $a_1 \cdot a_4 = 8 = 2^3$;
- $a_1 \cdot a_6 = 1 = 1^3$
- $a_2 \cdot a_3 = 27 = 3^3$;
- $a_3 \cdot a_5 = 216 = 6^3$;
- $a_4 \cdot a_6 = 8 = 2^3$.

E. Rock Is Push

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Вы находитесь в левой верхней клетке $(1,1)$ лабиринта размера $n \times m$ клеток. Ваша цель — попасть в правую нижнюю клетку (n,m) . Вы можете двигаться только вправо и вниз по одной клетке за ход. Переход вправо из клетки (x,y) ведёт в клетку $(x,y+1)$, а переход вниз — в клетку $(x+1,y)$.

В некоторых клетках лабиринта лежат камни. Когда вы перемещаетесь в клетку с камнем, камень сдвигается в следующую клетку по направлению движения. Если в следующей клетке также есть камень, он сдвигается дальше в том же направлении, и так далее.

Лабиринт окружён непроницаемыми стенами, поэтому ходы, в результате которых вы или камень выходят за границы лабиринта, невозможны.

Вычислите количество различных способов добраться от старта до цели по модулю 10^9+7 . Два способа считаются различными, если существует клетка, посещённая в одном из способов, но не посещённая в другом.

Входные данные

В первой строке записано два целых числа n, m — размеры лабиринта ($1 \leq n, m \leq 2000$).

Следующие n строк описывают лабиринт. Каждая из этих строк содержит m символов. j -й символ i -й из этих строк равен «R», если клетка (i,j) содержит камень, или «.», если клетка (i,j) пуста.

Гарантируется, что стартовая клетка $(1,1)$ пуста.

Выходные данные

Выведите одно целое число — количество различных способов добраться из клетки $(1,1)$ в клетку (n,m) по модулю 10^9+7

Пример

входные данные
1 1 .
выходные данные
1

входные данные
2 3R
выходные данные
0

ВХОДНЫЕ ДАННЫЕ
4 4
...R
.RR.
.RR.
R...
ВЫХОДНЫЕ ДАННЫЕ
4

Примечание

В первом примере нам нельзя (и не нужно) двигаться, поэтому единственный способ добраться посещает единственную клетку $(1,1)$.

Во втором примере цель заблокирована и в неё нельзя попасть.

Иллюстрации к третьему примеру можно найти

Здесь: <https://assets.codeforces.com/rounds/1225/index.html>

Г. Фабрика деревьев

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод
вывод: стандартный вывод

Байтландская фабрика деревьев производит деревья для всевозможных промышленных применений. Вам требуется оптимизировать построение дерева определённого типа для большого и важного заказа.

Нужное дерево является подвешенным деревом с n вершинами, и его вершины пронумерованы различными числами от 0 до $n-1$. Вершина с номером 0 является корнем дерева, и для любой некорневой вершины v номер её родителя $p(v)$ меньше номера v .

Все деревья на фабрике делаются из заготовок-бамбуков. *Бамбук* — это такое корневое дерево, что каждая вершина имеет ровно одного сына, кроме единственной вершины-листа, у которой нет детей. Вершины бамбука-заготовки могут быть пронумерованы как угодно до начала обработки.

Чтобы превратить бамбук в нужное дерево, вы можете применять один тип операций: выбрать произвольную некорневую вершину v , такую что её родитель $p(v)$ также не является корнем. Операция состоит в изменении родителя v на родителя его родителя $p(p(v))$. Родители всех остальных вершин остаются без изменений, в частности, поддерево v не меняется.

Эффективность крайне важна, поэтому вам требуется минимизировать количество операций для превращения бамбука-заготовки в нужное дерево. Постройте любую оптимальную последовательность операций.

Обратите внимание, что нумерация вершин в полученном дереве должна совпадать с нумерацией в данном дереве. Формально, номера корней должны совпадать, а также для некорневых вершин с одинаковыми номерами должны совпадать номера их родителей.

Гарантируется, что во всех тестах к этой задаче ответ существует, и, более того, в оптимальной последовательности не более 10^6 операций. Обратите внимание, что **любой взлом, не удовлетворяющий этим ограничениям, будет некорректным**.

Входные данные

В первой строке записано единственное целое число n — количество вершин дерева ($2 \leq n \leq 10^5$).

Во второй строке записано $n-1$ целое число $p(1), \dots, p(n-1)$ — номера родителей вершин $1, \dots, n-1$ соответственно ($0 \leq p(i) < i$).

Выходные данные

В первой строке выведите n различных целых чисел id_1, \dots, id_n — изначальная нумерация вершин бамбука-заготовки, начиная с корня ($0 \leq id_i < n$).

Во второй строке выведите одно целое число k — количество операций в вашей последовательности ($0 \leq k \leq 10^6$).

В третьей строке выведите k целых чисел v_1, \dots, v_k , описывающих операции по порядку. i -я операция состоит в изменении $p(v_i)$ на $p(p(v_i))$. Каждая операция должна быть корректной, т.е. v_i и $p(v_i)$ на момент операции не могут быть корнем.

Примеры

входные данные
5 0 0 1 1
выходные данные
0 2 1 4 3 2 1 3

входные данные
4 0 1 2
выходные данные
0 1 2 3 0

G. Получить 1

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

На доске написано n положительных чисел. Также выбрано положительное число $k \geq 2$, и ни одно из написанных чисел не делится на k . За одну операцию разрешается стереть с доски любые два числа x и y и записать вместо них $f(x+y)$, где $f(x)$ равно x , если x не делится на k , и $f(x) = f(x/k)$ в противном случае.

В конце на доске останется одно число. Возможно ли сделать это число равным 1 ? Если это так, восстановите любую последовательность операций, которая к этому приводит.

Входные данные

В первой строке записано два целых числа n и k — исходное количество чисел на доске и выбранное число ($2 \leq n \leq 16$, $2 \leq k \leq 2000$).

Во второй строке записано n положительных чисел a_1, \dots, a_n , изначально написанных на доске. Гарантируется, что ни одно из чисел a_i не делится на k , а также сумма всех a_i не превосходит 2000 .

Выходные данные

Если получить 1 в качестве последнего числа невозможно, выведите единственную строку «NO».

В противном случае, в первой строке выведите «YES», а затем выведите $n-1$ строку с описанием операций. i -я из этих строк должна содержать два целых числа x_i и y_i , которые необходимо заменить на $f(x_i+y_i)$ на i -й операции. Если существует несколько подходящих последовательностей, выведите любую из них.

Примеры

входные данные
2 2
1

ВЫХОДНЫЕ ДАННЫЕ
YES 1 1

ВХОДНЫЕ ДАННЫЕ
4 3 7 8 13 23
ВЫХОДНЫЕ ДАННЫЕ
YES 23 13 8 7 5 4

ВХОДНЫЕ ДАННЫЕ
3 4 1 2 3
ВЫХОДНЫЕ ДАННЫЕ
NO

Примечание

Во втором примере:

- $f(8+7)=f(15)=f(5)=5$;
- $f(23+13)=f(36)=f(12)=f(4)=4$;
- $f(5+4)=f(9)=f(3)=f(1)=1$.

