Задача A. Приготовление шашлыка

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

Больше всего в жизни Мирослав любит смотреть на горящий огонь, вкусно ужинать и участвовать в олимпиадах по информатике. Поэтому нет ничего удивительного в том, что в этой задаче речь пойдёт про шашлык — любимое блюдо Мирослава. Для приготовления шашлыка Мирослав использует металлическую коробку прямоугольной формы без верхней грани, которая называется мангал. В ней он сжигает несколько деревянных поленьев. Когда поленья прогорают и дно коробки покрывается тлеющими углями, можно начинать готовить шашлык. Для этого на верхнюю открытую грань мангала кладутся параллельно друг другу металлические палочки, называемые шампурами, на которые нанизаны нарезанные кусочки мяса, рыбы, грибов или овощей, в зависимости от настроения Мирослава в этот день. Искусство приготовления шашлыка не в последнюю очередь определяется умением правильно и своевременно переворачивать шампуры для равномерной прожарки блюда.

Итак, в этот раз Мирослав разложил параллельно друг другу n шампуров и пронумеровал их слева направо целыми числами от 1 до n. При этом для достижения наибольшего кулинарного эффекта Мирослав плотно прижимает шампуры друг к другу, так что если он переворачивает шампур с номером i, то это так же приводит к переворачиванию соседних шампуров с номерами $i-1$ и $i+1$ (если такие существуют). Они в свою очередь уже не переворачивают своих соседей, то есть если $n = 6$ и Мирослав перевернёт шампур с номером 3, то в итоге перевёрнутыми окажутся шампуры с номерами 2, 3 и 4. Если же после этого Мирослав перевернёт шампур с номером 1, то перевёрнутыми окажутся шампуры 1, 3 и 4, а шампур 2 вернётся в исходное положение, так как является соседом шампура 1.

Как мы уже упоминали выше, искусство приготовления шашлыка требует делать всё вовремя, так что Мирослав хочет научиться переворачивать все n шампуров за минимальное количество действий. Например, для рассмотренного выше примера $n = 6$ можно справиться за два действия, просто перевернув шампуры с номерами 2 и 5.

Помогите Мирославу перевернуть все n шампуров.

Формат входных данных

В первой строке входных данных записано целое число n ($1 \leqslant n \leqslant 50$) — количество шампуров, выложенных Мирославом на мангале.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество действий, которое потребуется совершить Мирославу, чтобы перевернуть все n шампуров. В каждой из последующих k строк выведите одно целое число от 1 до n, обозначающее номер шампура, который требуется перевернуть на соответствующем шаге.

Примеры

стандартный ввод	стандартный вывод
6	2 2 5
5	2 2 5

Замечание

Обратите внимание, что во втором примере правильным ответом также является перевернуть шампуры с номерами 1 и 4, но если перевернуть шампуры с номерами 2 и 4, то шампур номер 3 окажется в исходном состоянии.

Система оценки

В данной задаче 50 тестов, проверяющих все возможные значения параметра n в некотором произвольном порядке. Каждый тест, **включая тесты из условия**, стоит 2 балла и оценивается независимо.

Задача B. Разработка задач

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

Иннокентий неоднократно приходил на школьные олимпиады по информатике в качестве участника, а теперь стал студентом и принимает участие в работе жюри. В этот раз он отвечает за разработку следующей задачи.

Имеются две остановки автобусов A и B, и известно, что в течение дня n автобусов совершают поездку от A до B. От остановки A до остановки B самым быстрым образом можно доехать за время t, но можно ехать и дольше, при этом автобусы могут произвольным образом обгонять друг друга в процессе следования по маршруту.

На каждой из остановок можно прочитать упорядоченный список моментов времени, когда на эту остановку приходит автобус, обозначим как $a_1 < a_2 < \ldots < a_n$ список для остановки A и как $b_1 < b_2 < \ldots < b_n$ список для остановки B. Обозначим через $p_1, p_2, \ldots, p_n$ какую-нибудь перестановку чисел от 1 до n, то есть такую последовательность чисел, что каждое целое число от 1 до n встречается в ней ровно один раз. Будем считать, что эта перестановка задаёт сопоставление между порядком отхода автобусов от A и порядком их прихода в B, то есть автобус, ушедший с остановки A в момент времени a_1, должен прийти на остановку B в момент времени b_{p_1}, автобус, ушедший в момент времени a_2, должен прийти на остановку B в момент времени b_{p_2} и так далее. Сопоставление p_i будем считать допустимым, если ни для одного из автобусов не нарушено условие на время проезда, то есть $a_i + t \leqslant b_{p_i}$. При этом гарантируется, что тождественная перестановка (перестановка, в которой $p_i = i$) является допустимым сопоставлением автобусов в расписании.

Далее, от решающего задачу требуется определить, насколько сильно мог задержаться каждый из автобусов, то есть требуется независимо для каждого i найти такое максимальное значение $x_i \leqslant n$, что существует допустимый порядок прихода автобусов p_i, в котором i-й автобус приходит x_i-м, то есть $p_i = x_i$.

К несчастью для Иннокентия, процесс разработки олимпиадной задачи подразумевает подготовку набора тестов, так что теперь уже самому Иннокентию предстоит решить обратную к данной задачу. По заданным значениям количества автобусов n, временам отправления от первой остановки a_i, значению параметра t и значениям функции x_i, найти какое-нибудь подходящее расписание b_i, такое что четвёрка n, a_i, b_i и t определяет корректный ввод для исходной задачи, и ответом на эту задачу является последовательность x_i, или определить, что подходящей последовательности $b_1 < b_2 < \ldots b_n$ не существует.

Формат входных данных

В первой строке записаны два числа n и t ($1 \leqslant n \leqslant 200\,000$, $1 \leqslant t \leqslant 10^{18}$) — количество автобусов в расписании на обеих остановках и минимальное возможное время поездки автобуса от A до B.

Во второй строке записаны n целых чисел a_i ($1 \leqslant a_1 < a_2 < \ldots < a_n \leqslant 10^{18}$), определяющие времена отправления автобусов от остановки A.

В третьей строке записаны n чисел x_i ($1 \leqslant x_i \leqslant n$), i-е из которых означает максимально допустимую позицию в расписании автобусов на остановке B, которую мог занять автобус, отъехавший i-м от остановки A.

Формат выходных данных

Если решение существует, в первой строке выведите слово «`Yes`» (без кавычек). Во второй строке выведите n целых чисел b_i ($1 \leqslant b_1 < b_2 < \ldots < b_n \leqslant 3 \cdot 10^{18}$). Гарантируется, что если решение существует, то существует и решение, удовлетворяющее данному ограничению на значения b_i. Если правильных ответов несколько, то разрешается вывести любой из них.

Если подходящих значений b_i не существует, выведите слово «`No`» (без кавычек) в единственной строке вывода.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы и всех тестов **предыдущих** групп.

Группа	Баллы	Дополнительные ограничения			Комментарий
		n	t	a_i	
0	0	–	–	–	Тесты из условия.
1	10	$n = 2$	$t \leqslant 10^9$	$a_i \leqslant 10^9$	
2	20	$n = 3$	$t \leqslant 10^9$	$a_i \leqslant 10^9$	
3	40	$n \leqslant 5000$	$t \leqslant 10^9$	$a_i \leqslant 10^9$	
4	30	–	–	–	

Примеры

стандартный ввод	стандартный вывод

`3 10`
`4 6 8`
`2 2 3`
	`Yes`
	`16 17 21`

`2 1`
`1 2`
`2 1`
	`No`

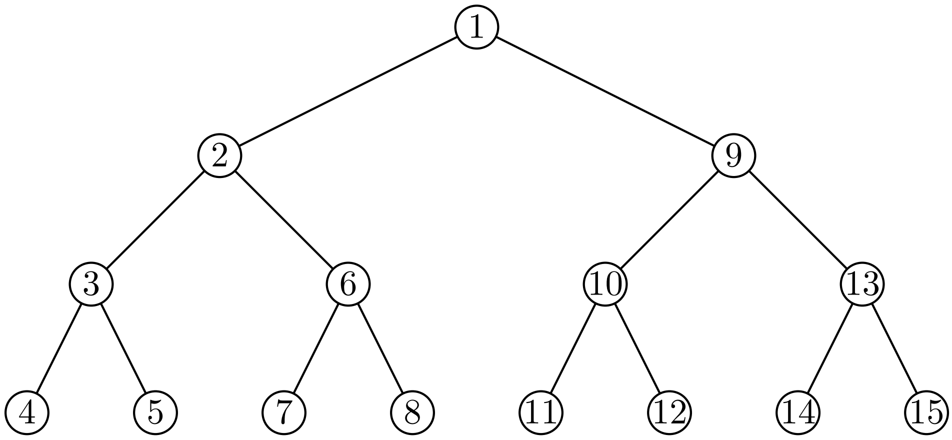
Задача C. Бинарное дерево

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

Назовём полным бинарным деревом порядка 1 дерево, состоящее из единственной вершины. Назовём полным бинарным деревом порядка $k > 1$ дерево, состоящее из вершины, называемой корнем дерева, у которой есть два сына (левый и правый), каждый из которых является корнем полного бинарного дерева порядка $k - 1$.

Рассмотрим следующий алгоритм нумерации вершин полного бинарного дерева. Присвоим минимальный незанятый положительный целый номер корню дерева, затем запустим алгоритм для левого поддерева дерева, затем запустим алгоритм для правого поддерева.

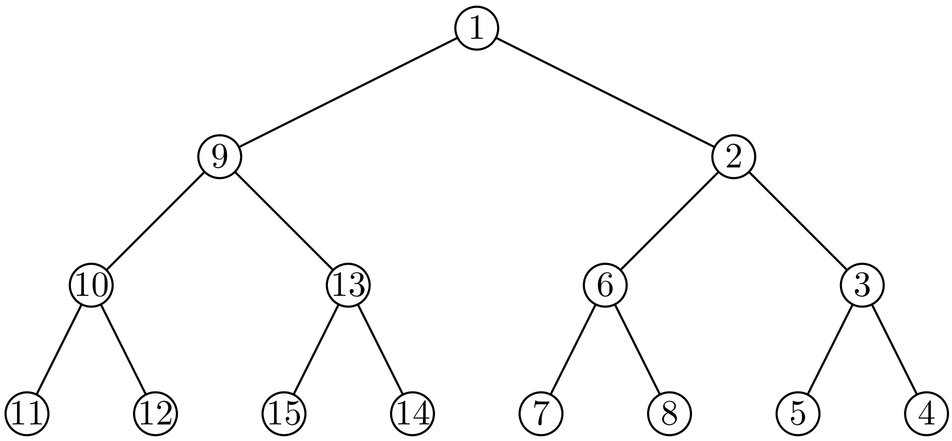
Пример нумерации для полного бинарного дерева порядка 4.



В полном бинарном дереве порядка k отметили n вершин, i-я отмеченная вершина имела номер a_i в дереве. С деревом можно произвести несколько (возможно ноль) операций следующего вида:

Для некоторой вершины v поменять порядок сыновей. В результате этой операции левый сын становится правым, а правый — левым, при этом структура вершин в каждом из поддеревьев её сыновей не меняется.

Например, в следующем бинарном дереве эту операцию применили к вершинам 1, 2, 3 и 13.



После выполнения всех операций вершины дерева заново нумеруются. Для i-й отмеченной вершины имеется условие, что после перенумерации она имеет номер b_i. Определите, возможно ли так произвести с деревом описанные выше операции, чтобы после перенумерации условия для всех отмеченных вершин оказались выполненными одновременно.

Формат входных данных

В первой строке заданы два целых числа n и k ($1 \leqslant n \leqslant \min(300\,000, 2^k - 1)$, $1 \leqslant k \leqslant 60$) — число отмеченных вершин и глубина полного бинарного дерева соответственно.

В следующих n строках заданы пары целых чисел a_i и b_i ($1 \leqslant a_i, b_i \leqslant 2^k - 1$) — исходный и требуемый номера i-й отмеченной вершины соответственно.

Гарантируется, что все a_i различны, и все b_i различны.

Формат выходных данных

Если существует такой набор операций, что после их применения каждая из n отмеченных вершин получит требуемый новый номер (вершина, исходно имевшая номер a_i, получит номер b_i), выведите «`Yes`» (без кавычек), иначе выведите «`No`» (без кавычек).

Примеры

стандартный ввод	стандартный вывод

`3 3`
`2 5`
`7 3`
`1 1`
	`Yes`

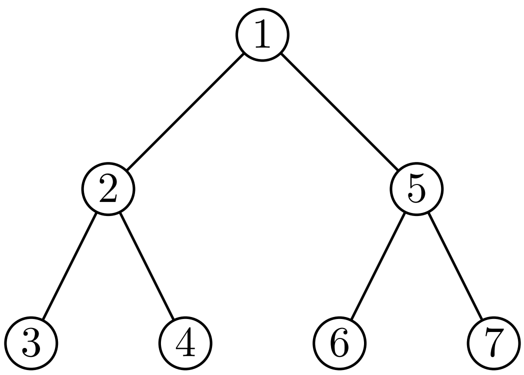
`2 3`
`2 5`
`3 4`
	`No`

`1 2` `1 3`	`No`

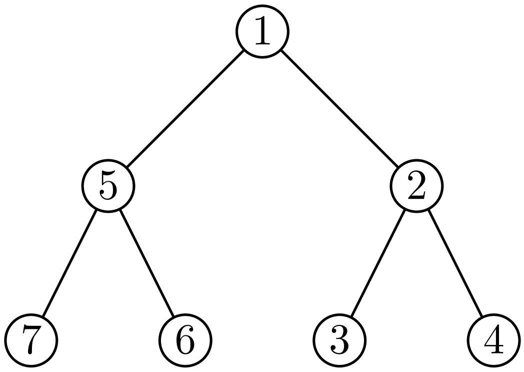
`6 4`
`4 15`
`15 7`
`7 11`
`8 12`
`6 10`
`10 3`
	`Yes`

Замечание

В первом примере исходная нумерация вершин дерева такова:



Одним из вариантов достичь требуемой нумерации является изменение порядка сыновей у вершин 1 и 5.



Во втором примере мы обязаны поменять порядок сыновей у корня, чтобы вершина 2 имела номер 5 после перенумерации. В этом случае вершина 3 может иметь только номера 6 и 7 после перенумерации.

В третьем примере мы можем менять порядок сыновей только у корня дерева, и, сколько бы раз мы его ни меняли, корень дерева не изменит своего номера.

Один из вариантов достичь требуемой нумерации в четвертом примере показан в условии задачи.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы и всех групп, от которых зависит данная группа.

Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n	k		
0	0	–	–	–	Тесты из условия
1	10	$n \leqslant 31$	$k \leqslant 5$	0	–
2	20	–	$k \leqslant 20$	0 – 1	–
3	10	$n = 1$	–	–	–
4	10	$n \leqslant 2$	–	3	–
5	15	$n \leqslant 5000$	–	0 – 1, 3 – 4	–
6	25	$n \leqslant 50\,000$	–	0 – 1, 3 – 5	–
7	10	–	–	0 – 6	–

Задача D. Летняя Выставка Энотер

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Некоторые люди обожают решать контесты по спортивному программированию, Дина же увлекается фотографией. Как только Байтландский ботанический сад объявил о старте Летней выставки энотер, она решила попробовать свою новую камеру в действии.

На выставке представлены $l = 10^{100}$ энотер, выставленных в ряд, и пронумерованных от 0 до $l - 1$. Объектив камеры позволяет сделать снимок с ровно w цветами, то есть Дина может снять любую фотографию, содержащую цветы с номерами с x по $x + w - 1$ для некоторого целого x от 0 до $l - w$. Такую фотографию мы будем обозначать как $[x, x + w - 1]$.

Во время выставки она сделала n фотографий, i-я по порядку фотография в наших обозначениях выглядит как $[x_i, x_i + w - 1]$. Увидев, что вечером энотеры раскрываются, она решила сделать видео с замедленной съемкой (time-lapse) из этих фотографий.

Процесс заключается в следующем: Дина обрезает фотографии, оставляя от каждой из них подотрезок, состоящий ровно из k цветов, составляет из них видео, сохраняя порядок их следования, и вуаля — произведение искусства готово!

Назовем сценой непрерывную последовательность фотографий такую, что множество цветов на них совпадает. Изменение множества цветов между двумя сценами мы назовем *склейкой*. Например, пусть на первой фотографии запечатлены цветы $[1, 5]$, на второй — $[3, 7]$, а на третьей — $[8, 12]$. Если $k = 3$, то можно вырезать на первой и второй фотографии отрезок цветов $[3, 5]$, а на третьей фотографии — отрезок цветов $[9, 11]$. Тогда первые две фотографии образуют одну сцену, третья фотография образует вторую сцену, а изменение между этими сценами, которое происходит между второй и третьей фотографией — склейка. Если же $k = 4$, то каждый из двух переходов между фотографиями вынужден быть склейкой.

Дина хочет, чтобы количество склеек на итоговом видео было минимально. Помогите ей! Посчитайте минимальное количество склеек для разных значений k.

Формат входных данных

Первая строка содержит три положительных целых числа n, w, q ($1 \leqslant n, q \leqslant 100\,000$, $1 \leqslant w \leqslant 10^9$) — количество сделанных фотографий, количество цветов на одной фотографии и количество запросов соответственно.

Следующая строка содержит n неотрицательных чисел x_i ($0 \leqslant x_i \leqslant 10^9$) — номера самых левых энотер на каждой из фотографий.

Следующая строка содержит q положительных чисел k_i ($1 \leqslant k_i \leqslant w$) — значения k, при которых нужно найти ответ на задачу.

Гарантируется, что все k_i различны.

Формат выходных данных

Выведите q чисел — для каждой итоговой ширины обрезанной фотографии k_i определите минимальное количество склеек, которого можно добиться.

Примеры

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

Замечание

Система оценки

Тесты по данной задаче состоят из четырех групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **зависимых** групп кроме тестов из условия.

Группа	Тесты	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
			n	q		
0	1 – 2	0	–	–	–	Тесты из условия
1	3 – 17	15	$n \leqslant 50$	$q \leqslant 50$	–	$\lvert x_i - x_{i-1} \rvert \leqslant 1$
2	18 – 55	20	$n \leqslant 10\,000$	$q \leqslant 10\,000$	1	
3	56 – 72	25	$n \leqslant 100\,000$	$q \leqslant 100\,000$	1	$\lvert x_i - x_{i-1} \rvert \leqslant 1$
4	73 – 101	40	$n \leqslant 100\,000$	$q \leqslant 100\,000$	1, 2, 3	