Задача A. Разбиение на пары

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

Дана последовательность, состоящая из n целых чисел. Проверьте, можно ли разбить числа на пары таким образом, чтобы сумма чисел в каждой паре была четной.

Формат входных данных

В первой строке вводится число n ($1 \leqslant n \leqslant 10^3$) — количество элементов в последовательности.
Во второй строке вводится n целых чисел a_i ($1 \leqslant a_i \leqslant 10^5$) — элементы последовательности.

Формат выходных данных

Выведите «Yes», если элементы последовательности можно разбить на пары с четной суммой и «No» в противном случае.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4	Yes
3 3 1 2	No
6 6 5 1 3 2 4	No
2 11 99	Yes

Замечание

В первом примере числа можно разбить на пары $1 + 3$ и $2 + 4$.

Во втором примере числа нельзя разбить на пары с четной суммой, т.к. количество чисел нечетное.

В третьем примере не существует разбиения на пары с четной суммой.

В четвертом примере единственная пара имеет четную сумму.

Задача B. Тысячный день

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

Сегодня Слава тысячный день подряд готовится к олимпиаде, решая задачи. На сайте для подготовки к олимпиадам для каждой задачи указана ее сложность, которая обозначается числом от 1 до N. К сожалению, произошел сбой и сложность задач перестала отображаться, но прочитав задачу, Слава может определить её сложность.

Известно, что задач каждой сложности осталось ровно K штук. Задачи первого уровня сложности Слава решает за A минут, задачи второго уровня сложности за $2 \cdot A$ минут, ..., задачи сложности N он решает за $N \cdot A$ минут.

Слава хочет решить одну задачу. До конца дня осталось T минут (при этом Слава может отправлять решение в последнюю минуту). На прочтение одной задачи он всегда тратит M минут и после этого может либо решить задачу, либо перейти к следующей. Сможет ли Слава до конца дня решить хотя бы одну задачу независимо от порядка их чтения?

Формат входных данных

Первая строка входных данных содержит целое число N ($1 \leqslant N \leqslant 10^6$) — количество видов сложностей задач.

Вторая строка содержит целое число K ($1 \leqslant K \leqslant 10^6$) — количество задач каждой сложности.

Третья строка содержит целое число A ($1 \leqslant A \leqslant 10^6$) — время, за которое Слава решает самую легкую задачу.

Четвёртая строка содержит целое число M ($1 \leqslant M \leqslant 10^6$) — время, за которое Слава может прочитать условие и оценить его сложность.

Пятая строка содержит целое число T ($1 \leqslant T \leqslant 10^{18}$) — оставшееся время до конца дня.

Формат выходных данных

Если Слава точно успеет решить 1 задачу до конца дня, то выведите «Yes» (без кавычек), иначе выведите «NO» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
2 1 10 3 16	Yes
3 1 10 1 11	No

Замечание

В первом тесте всего 2 задачи. В худшем случае Славе сначала попадется сложная задача, а потом простая. На чтение условия сложной задачи он потратит 3 минуты и примет решение не решать эту задачу, а в простой задаче он прочитает условие за 3 минуты и решит ее за 10 минут. На обе задачи он потратит 16 минут, что не превосходит времени до конца дня.

Во втором тесте Слава успевает, только если первая задача попадется самая лёгкая. Т.к. ему может не повезти, то он не гарантированно успеет и ответ No.

Задача C. Квадратная карта

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

Есть таблица размера $n \times n$, в клетках которой записаны числа от 1 до n.

Каждая клетка таблицы должна быть раскрашена в один из c цветов, при этом все клетки с одинаковым числом должны быть покрашены в один и тот же цвет. Для любых двух соседних по стороне клеток таблицы, в которых записаны различные числа, цвета этих клеток должны быть различны. Разноцветностью таблицы назовём минимальное число c, для которого есть корректная раскраска таблицы в c цветов.

Выведите таблицу размера $n \times n$, в клетках которой записаны числа от 1 до n, с максимальной разноцветностью. Каждое число может встречаться в таблице любое число раз.

Формат входных данных

В единственной строке ввода содержится одно число n ($2 \leqslant n \leqslant 500$) — сторона таблицы.

Формат выходных данных

Выведите таблицу размера $n \times n$ с максимальной разноцветностью. Если существует несколько ответов, вы можете вывести любой из них.

Примеры

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

Замечание

Разноцветность во втором примере равна 4. В этой таблице никакие два различных числа нельзя покрасить в один цвет, т.к. для каждой пары чисел найдутся две соседние по стороне клетки с этими числами.

Задача D. Среднее не найдено

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

Дан массив A длины n и массив B длины m. Найдите такую пару индексов (i, j) $(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$, что $\frac{A_i+B_j}{2}$ не принадлежит ни одному из массивов.

Формат входных данных

Первая строка содержит одно число n $(1 \leqslant n \leqslant 2 \cdot 10^5)$ — длину массива A.
Вторая строка содержит n целых чисел $(0 \leqslant A_i \leqslant 10^9)$ — элементы массива A.
Третья строка содержит одно число m $(1 \leqslant m \leqslant 2 \cdot 10^5)$ — длину массива B.
Четвёртая строка содержит m целых чисел $(0 \leqslant B_i \leqslant 10^9)$ — элементы массива B.

Формат выходных данных

В единственной строке выведите 2 индекса i и j $(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$ такие, что $\frac{A_i+B_j}{2}$ не принадлежит ни одному из массивов. Если такой пары индексов нет, выведите одно число -1.

Примеры

стандартный ввод	стандартный вывод
4 2 3 4 6 2 2 3	1 2
3 1 1 1 4 1 1 1 1	-1
2 2 4 1 6	2 1

Замечание

В первом примере $\frac{a_1+b_2}{2} = \frac{2+3}{2} = 2.5$. Такого числа нет ни в одном из массивов.

Задача E. ABC

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

Дан набор строк $s_1, s_2, \ldots, s_n$. Вы можете как угодно переставить эти строки, после чего записать конкатенацию этих строк в строку t.

Напомним, что конкатенацией строк a и b называется строка, полученная дописыванием строки b к строке a. $a + b = a_1 a_2 \ldots a_p b_1 b_2 \ldots b_q$, где p и q — длины строк a и b соответственно.

Другими словами, вы можете составить строку $t = s_{p_1} + s_{p_2} + \ldots + s_{p_n}$, где $p_1, p_2, \ldots, p_n$ — различные целые числа от 1 до n.

Определите, можно ли таким образом составить строку t, чтобы она не содержала подпоследовательность «`abc`», и если можно, то в каком порядке нужно записать строки из набора. То есть **не должно существовать** таких индексов i, j, k, таких что $1 \leqslant i < j < k \leqslant |t|$ и $t_i =$ `'a'`, $t_j =$ `'b'` и $t_k =$ `'c'`.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 10^5$) — количество строк в наборе.

i-я из следующих n строк содержит одну строку s_i ($1 \leqslant |s_i| \leqslant 2 \cdot 10^5$), состоящую из только из символов `'a'`, `'b'`, `'c'`, — i-я строка из набора.

Гарантируется, что сумма длин всех строк в наборе не превосходит $2 \cdot 10^5$.

Формат выходных данных

Если искомую строку t можно составить, выведите в первой строке «`Yes`» (без кавычек). Далее, во второй строке выведите n различных целых чисел $p_1, p_2, \ldots, p_n$ ($1 \leqslant p_i \leqslant n$) — номера строк из набора, такие что $t = s_{p_1} + s_{p_2} + \ldots + s_{p_n}$. Если существует несколько возможных решений, выведите любое из них.

Если искомую строку t невозможно составить, выведите в единственной строке «`No`» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
1 ccccbbbbaaa	Yes 1
5 a bca bb ca caba	Yes 3 2 4 5 1
2 cab acba	No

Замечание

В первом примере строка $t = s_1 =$ «`ccccbbbbaaa`» не содержит подпоследовательность «`abc`».

Во втором примере строка $t = s_3 + s_2 + s_4 + s_5 + s_1 =$ «`bbbcacacabaa`» не содержит подпоследовательность «`abc`».

В третьем примере либо $t = s_1 + s_2 =$ «`cabacba`» (содержит подпоследовательность «`abc`», так $t_2 =$ `'a'`, $t_3 =$ `'b'`, $t_5 =$ `'c'`), либо $t = s_2 + s_1 =$ «`acbacab`» (содержит подпоследовательность «`abc`», так $t_1 =$ `'a'`, $t_3 =$ `'b'`, $t_5 =$ `'c'`).

Задача F. Циклическая сумма

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

Даны два массива a и b длины n. Также есть массив c длины n, изначально заполненный нулями. Далее, каждую секунду происходит следующее:

- К массиву c поэлементно прибавляется массив b. Другими словами, для всех i от 1 до n выполняется присваивание $c_i = c_i + b_i$.
- Массив b циклически сдвигается на 1 влево. То есть, если $b = [b_1, b_2, \dots b_n]$, то он станет равным $[b_2, b_3, \dots, b_{n-1}, b_1]$.

Например, пусть b изначально равен $[1, 3, 2, 2]$. Тогда через 1 секунду $c = [1, 3, 2, 2]$, а $b = [3, 2, 2, 1]$. Через 2 секунды $c = [4, 5, 4, 3]$, а $b = [2, 2, 1, 3]$. Через 3 секунды $c = [6, 7, 5, 6]$, а $b = [2, 1, 3, 2]$. Обратите внимание, что мы каждый раз прибавляем к массиву c циклически сдвинутый на предыдущей секунде массив b.

Определите минимальное количество секунд, через которое массив c станет равен массиву a. Если такого никогда не произойдёт, то выведите -1. Обратите внимание, что **ответ может быть равен** 0.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 1500$) — длина массивов.

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($-10^9 \leqslant a_i \leqslant 10^9$) — элементы массива a.

Третья строка содержит n целых чисел $b_1, b_2, \dots, b_n$ ($-10^9 \leqslant b_i \leqslant 10^9$) — элементы массива b.

Формат выходных данных

Выведите одно целое число — минимальное количество секунд, через которое массив c станет равным массиву a. Если такого никогда не произойдёт, то выведите -1.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1 2 3	1
4 6 7 5 6 1 3 2 2	3
1 0 0	0
3 1 1 1 -1 -1 -1	-1

Замечание

В первом примере массив b изначально равен $[1, 2, 3]$. Через 1 секунду массив $c = [1, 2, 3]$, что равно массиву a. Поэтому ответ равен 1.

Второй пример описан выше в условии задачи.

В третьем примере массив c уже изначально равен массиву a, поэтому ответ равен 0.

В четвёртом примере массив c никогда не станет равен массиву a.

Задача G. Последовательные увеличения

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

Вам дан массив a из n целых чисел — $a_1, a_2, a_3, \ldots, a_n$.
Вы можете произвольное число раз выполнить следующую операцию:

- выбрать индекс i $(1 \leqslant i \leqslant n-1)$, такой что $a_i = a_{i+1}$.
- затем увеличить элемент с индексом $i+1$ на 1: $a_{i+1} = a_{i+1} + 1$.

Определите, какую максимальную сумму массива a можно получить после выполнения таких операций.

Формат входных данных

В первой строке содержится единственное целое число n $(1 \leqslant n \leqslant 2 \cdot 10^5)$ — длина массива a.
Вторая строка содержит n целых чисел $a_1, a_2, a_3, \ldots, a_n$ $(1 \leqslant a_i \leqslant 10^9)$ — элементы массива a.

Формат выходных данных

Выведите единственное число — максимальную сумму массива a, которую можно получить, выполняя операции.

Примеры

стандартный ввод	стандартный вывод
3 1 1 1	6
5 1 2 3 4 5	15
6 1 1 3 2 2 5	16

Замечание

В первом примере, можно проделать следующую последовательность операций:

1. выбрать $i = 2$, затем увеличить a_3 на 1. Таким образом, массив a станет равен $[1, 1, 2]$.
2. выбрать $i = 1$, затем увеличить a_2 на 1. Таким образом, массив a станет равен $[1, 2, 2]$.
3. выбрать $i = 2$, затем увеличить a_3 на 1. Таким образом, массив a станет равен $[1, 2, 3]$.

Итого, сумма массива a станет равна 6.

Задача N. Паскаль

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

Однажды Никита встретил Паскаля, который показал ему свою новейшую разработку — треугольник Паскаля.

Треугольник состоит из бесконечного количество строк, в которых записаны числа. В первой строке треугольника записано число 1. В каждой следующей строке количество чисел увеличивается на единицу, и каждое число равно сумме двух его соседей сверху (кроме двух крайних чисел — они равны своему единственному верхнему соседу). Ниже приведен пример первых шести строк треугольника Паскаля:

					1					
				1		1				
			1		2		1			
		1		3		3		1		
	1		4		6		4		1	
1		5		10		10		5		1

Никита в свою очередь придумал задачу про треугольник Паскаля. Он предлагает вам найти строку с минимальным номером, которая содержит число n. Справитесь ли вы с этой задачей?

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 10^9$) — число, которое хочет найти Никита.

Формат выходных данных

Выведите минимальный номер строки, которая содержит число n.

Гарантируется, что ответ всегда существует.

Примеры

стандартный ввод	стандартный вывод
1	1
6	5
10	6
17	18

Замечание

В первом примере требуется найти первую строку, содержащую 1. Первая строка треугольника Паскаля содержит единственное число 1, поэтому ответ на первый пример 1.

Во втором примере требуется найти первую строку, сожержащую число 6. По рисунку выше можно заметить, что первая строка содержащая число 6 это 5 строка.

Задача I. Штрих-код

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

Иннокентий работает кассиром в магазине, он часто использует сканер штрих-кодов. Пусть штрих-код любимого батончика Иннокентия «WilkyMay», состоит из m полосок: полоска i имеет тип b_i.

Сканер считает два штрих-кода одинковыми, в случае, если можно переставить полоски в первом коде так, чтобы получился второй код. Два штрих-кода считаются почти одинковыми, если можно поменять тип одной из полосок в первом коде так, чтобы он стал одинаковым со вторым.



Принтер штрих-кодов выдал один длинный лист бумаги.

Иннокентию стало интересно, сколько существует способов вырезать из этой бумаги один штрих-код, который будет почти одинаковым с кодом «WilkyMay».

Формат входных данных

В первой строке дается три числа $n, m, k \quad (1 \leqslant m \leqslant n \leqslant 2 \cdot 10^5, 1 \leqslant k \leqslant 10^5)$;

Во второй строке дано описание полосок в распечатанном листке. Это n чисел $a_i \quad (1 \leqslant i \leqslant n, 1 \leqslant a_i \leqslant k)$;

В третьей строке описан штрих-код батончика «WilkyMay»: m чисел $b_i \quad (1 \leqslant i \leqslant m, 1 \leqslant b_i \leqslant k)$.

Формат выходных данных

Одно число — возможное количество способов вырезать из распечатанного листка штрих-код почти одинаковый с кодом «WilkyMay».

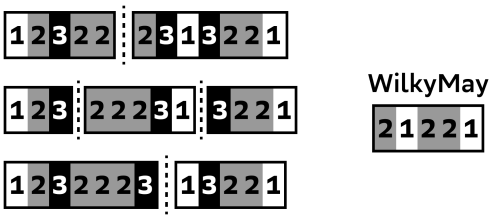
Примеры

стандартный ввод	стандартный вывод
5 5 3 3 1 3 1 3 3 1 2 3 1	1
12 5 3 1 2 3 2 2 2 3 1 3 2 2 1 2 1 2 2 1	3

Замечание

В первом примере сам распечатанный листок можно использовать как штрих-код, почти одинаковый с «WilkyMay». Изображение с подробным сопоставлением штрих-кода и набора полосок можно найти выше в условии

Во втором примере изображен штрих-код «WilkyMay» и все способы вырезать почти одинаковый с ним штрих код из распечатанного листка



Задача J. Повышение сотрудников

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

В подразделении компании работает n сотрудников, занумерованных числами от 1 до n. Руководитель подразделения имеет номер 1. У каждого сотрудника, кроме руководителя подразделения, есть ровно один непосредственный руководитель из числа сотрудников подразделения. Каждый сотрудник обладает единственной числовой характеристикой — компетентностью, для i-го сотрудника она равна c_i.

В ближайшее время планируется провести k переводов из подразделения в центральный офис. Каждый перевод происходит так: текущий руководитель подразделения переходит в центральный офис, его место занимает один из непосредственных подчиненных. В свою очередь, на место повышенного подчиненного назначается один из его непосредственных подчиненных и так далее.

HR-менеджер хотел бы определить, какую максимальную суммарную компетентность могут иметь k сотрудников, переведенных из подразделения в центральный офис.

Формат входных данных

В первой строке даны два числа n, k ($1 \leqslant k \leqslant n \leqslant 5 \cdot 10^3$) —количество сотрудников в подразделении и количество переводов в центральный офис соответственно.

Во второй строке содержатся n целых чисел $c_1, c_2, \ldots, c_n$ ($1 \leqslant c_i \leqslant 10^9$) —компетентность сотрудников.

В третьей строке задано $n-1$ число $p_2, p_3, \ldots, p_n$ ($1 \leqslant p_i < i$). Число p_i означает, что сотрудник с номером i является непосредственным подчиненным сотрудника с номером p_i. Обратите внимание, что нумерация начинается с 2, т.к. сотрудник 1 не имеет непосредственного руководителя.

Формат выходных данных

Выведите единственное число — максимальную суммарную компетентность сотрудников, переведенных в центральный офис.

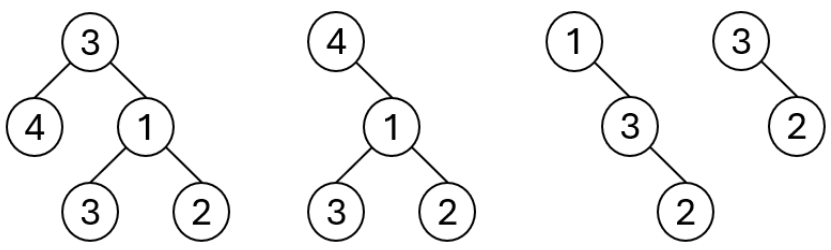
Примеры

стандартный ввод	стандартный вывод
3 2 1 2 3 1 1	4
5 3 3 4 1 3 2 1 1 3 3	8

Замечание

В первом примере у руководителя два непосредственных подчиненных с номером 2 и 3 и компетентностью 2 и 3 соответственно. При первом переходе к суммарной компетентности прибавляется 1 (компетентность руководителя) место руководителя занимает подчиненный с компетентностью 3. При следующем переходе к суммарной компетентности прибавляется 3, а единственным оставшимся сотрудником и новым руководителем подразделения оказывается сотрудник с компетентностью 2.

Изменение структуры подразделения во втором примере показано на рисунке:



Задача K. Логистика

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

Один известный маркетплейс решил оптимизировать свою логистику. Для этого они хотят минимизировать время загрузки товара на склады. Чтобы решить эту задачу, они обратились к вам за помощью!

Вам предстоит минимизировать это время для t разных складов независимо. Склад задается числом k — количеством доступных в нём ворот для подъезда фуры с товарами. Также известно, что на этот склад в течение дня должны прибыть n фур с товаром. В связи с разной грузоподъёмностью и другими техническими факторами, фуры имеют ограничения на ворота, к которым они могут подъехать и разгрузить свой товар. Конкретнее, i-й фуре допускается подъехать к одним из m_i ворот, задаваемых набором $a_{i,1}, a_{i,2}, \ldots, a_{i,m_i}$.

Чтобы наладить график поставок, было решено, что будет выбрано несколько разгрузочных часов в течение дня, в каждый из которых будет приезжать какое-то множество из запланированных фур на **разные** ворота, чтобы не мешать друг другу. Покупатели очень важны, поэтому товар не должен быть утерян. В связи с этим каждая фура должна приехать на склад ровно один раз в течение дня.

Так как скорость очень важна, вам необходимо, зная информацию о складе и фурах, составить график приезда всех фур на склад, минимизируя количество разгрузочных часов.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число t ($1 \leqslant t \leqslant 200$) — количество наборов входных данных. Далее следует описание наборов входных данных.

В первой строке каждого набора входных данных находится два целых числа k, n ($1 \leqslant k, n \leqslant 300$) — количество доступных ворот на складе и количество фур.

Следующие n строк каждого набора входных данных описывают ограничения для фур. В i-й из этих строк содержится число m_i ($1 \leqslant m_i \leqslant k$) — количество доступных ворот для i-й фуры. За которым следуют m_i различных чисел $a_{i,1}, a_{i,2}, \ldots, a_{i,m_i}$ ($1 \leqslant a_{i,j} \leqslant k$) — номера доступных ворот.

Гарантируется, что сумма всех m_i в одном тестовом наборе не превосходит 900.

Обозначим за N, K, M сумму n, k и m_i по всем тестовым наборам. Гарантируется, что $N, M, K \leqslant 900$.

Формат выходных данных

Для каждого набора входных данных сначала выведите одно целое число l — минимальное необходимое количество разгрузочных часов.

Затем выведите описание каждого из l разгрузочных часов.

Для i-го часа сначала выведите одно целое число c_i — число фур разгружаемых в этот час.

После выведите c_i строк, каждая из которых содержит два целых числа f, g ($1 \leqslant f \leqslant n, 1 \leqslant g \leqslant k$) — номер фуры и ворота, на которые она разгружается.

Необходимо, чтобы каждая из фур была разгружена ровно один раз в одни из **допустимых** ворот.

Пример

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