

Задача A. Automat

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

В одном из вузов Бйатландии стоит торговый автомат, продающий булочки. Автомат рассчитан на ассортимент из n видов булочек, пронумерованных от 1 до n . Для каждой булочки известна её цена. Чтобы купить булочку, необходимо выбрать код её вида и внести соответствующую сумму в автомат. При этом, если булочки этого вида в автомате закончились, автомат сообщает, что заказ невозможно выполнить и к приёму денег не переходит.

Хитрые студенты «подкрутили» автомат так, чтобы при выборе булочки i -го вида из автомата выпадало также по одной булочке каждого из видов $1, 2, \dots, i-1$ в случае, если булочки соответствующего вида в автомате не закончились (например, в случае, если в автомате есть по две булочки вида 1, 2, 4, 5, 7 и выбран вид 5, студент получает по одной булочке видов 1, 2, 4, 5; при выборе вида 6 автомат — как и в случае, если бы он был исправен — сообщит, что товар закончился).

Приехавший на стажировку студент Вася хочет потратить не более k бйатландских тугриков на покупку булочек в автомате так, чтобы суммарная стоимость получившейся покупки была максимальна.

Формат входных данных

В первой строке входного файла заданы два целых числа n и k ($1 \leq n \leq 40, 1 \leq k \leq 64\,000$) — максимальное количество сортов булочек в автомате и сумма, которой располагает Вася. Вторая строка содержит n целых чисел c_1, \dots, c_n ($1 \leq c_i \leq 40$). i -е из этих чисел задаёт цену булочки i -го вида. Третья строка содержит n целых чисел l_1, \dots, l_n ($0 \leq l_i \leq 40$) — количество булочек каждого вида в автомате перед первым заказом Васи.

Формат выходных данных

Выведите одно целое число — наибольшую суммарную стоимость булочек, извлечённых из автомата Васей.

Пример

standard input	standard output
6 8	30
7 2 3 5 7 2	
1 3 0 3 2 1	

Замечание

В данном примере Вася сначала заказывает булочку шестого вида, получая по одной булочке видов 1,2,4,5,6. Затем он заказывает булочку четвёртого вида, получая ещё по одной булочке видов 4 и 2.

Задача В. Touristic Bureau

Имя входного файла:	standard input
Имя выходного файла:	standard output
Ограничение по времени:	5 seconds
Ограничение по памяти:	256 mebibytes

В Байтбурге, втором по величине городе Байтландии, улицы направлены или с запада на восток, или с севера на юг. При этом расстояние между каждыми двумя соседними улицами одинаково и равно одному километру.

На некоторых перекрёстках Байтбурга расположены достопримечательности. При этом у каждой достопримечательности существует «коэффициент интереса», определённый в результате независимых исследований и обозначающий, насколько данная достопримечательность интересна туристам. Чем выше коэффициент, тем больший интерес вызывает достопримечательность.

Туристическое бюро «Bytebus» планирует организовать для гостей Байтбурга экскурсию на двухэтажном автобусе с посещением некоторых достопримечательностей. При этом для сохранения интереса на протяжении всей экскурсии коэффициенты интереса посещённых достопримечательностей должны строго возрастать.

При этом с каждого километра, который автобус провёл в пути, бюро получает с каждого туриста один байтладнский тугрик дохода. Кроме этого, посещение некоторых достопримечательностей является платным и тоже приносит бюро соответствующий доход.

При этом, следуя от одной достопримечательности к другой, водитель автобуса всегда выбирает кратчайший маршрут, проходящий по улицам Байтбурга; проезд мимо достопримечательности её посещением не считается.

Компания хочет организовать маршрут таким образом, чтобы доход компании с маршрута был максимальным.

Формат входных данных

В первой строке входного файла заданы два целых числа n и m ($2 \leq n, m \leq 1000$) — соответственно количество улиц Байтбурга, идущих с запада на восток, и улиц, идущих с севера на юг.

В последующих n строках, каждая из которых содержит m целых чисел, заданы коэффициенты интереса для соответствующих достопримечательностей. j -е число в i -й строке равно 0, если на перекрёстке i -й улицы, идущей с запада на восток, и j -й улицы, идущей с севера на юг, нет достопримечательности, и коэффициенту $w_{i,j}$ ($1 \leq w_{i,j} \leq 10^6$) в случае, если достопримечательность там присутствует. Гарантируется, что как минимум одна достопримечательность в Байтбурге есть.

В каждой из последующих n строк содержатся по m целых чисел $c_{i,j}$ ($0 \leq c_{i,j} \leq 10^9$). Число $c_{i,j}$, или j -е число в i -й строке, обозначает доход, который бюро получает с туриста за посещение достопримечательности на соответствующем перекрёстке. В случае, если достопримечательность отсутствует, $c_{i,j} = 0$.

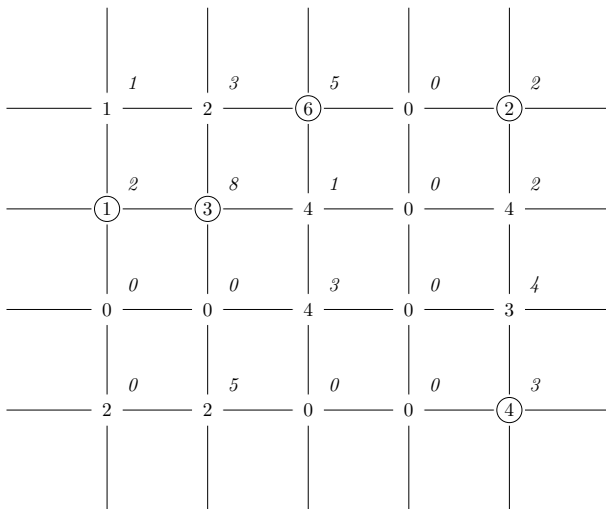
Формат выходных данных

Выведите одно число — наибольший доход с одного туриста, который компания может получить при организации экскурсии по описанным в условии правилам.

Пример

standard input	standard output
4 5 1 2 6 0 2 1 3 4 0 4 0 0 4 0 3 2 2 0 0 4 1 3 5 0 2 2 8 1 0 2 0 0 3 0 4 0 5 0 0 3	39

Замечание



На иллюстрации к примеру из условия обычным шрифтом обозначен «коэффициент интереса», курсивом — доходы Bytebus за посещение соответствующей достопримечательности. Достопримечательности, расположенные на наиболее доходном маршруте, обведены кружками. Соответственно, за посещение достопримечательностей бюро собирает 2, 2, 8, 3 и 5 тугриков соответственно, а за проезд — 19 тугриков.

Задача C. Sequence

Имя входного файла: standard input
Имя выходного файла: standard output
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Назовём последовательность целых чисел *k-чётной*, если сумма любых *k* идущих подряд элементов этой последовательности чётна.

Для данной последовательности найдите минимальное количество элементов, которое нужно изменить для того, чтобы последовательность стала *k-чётной*.

Формат входных данных

В первой строке входного файла заданы два целых числа n и k ($1 \leq k \leq n \leq 10^6$) — длина последовательности и параметр чётности. Вторая строка содержит n целых чисел a_i — элементы последовательности, перечисленные по возрастанию индекса ($0 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — минимальное количество элементов, которые требуется изменить для того, чтобы последовательность стала *k-чётной*.

Примеры

standard input	standard output
8 3 1 2 3 4 5 6 7 8	3
8 3 2 4 2 4 2 4 2 4	0

Задача D. DNA

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Байтландские учёные решили вывести с помощью генных модификаций новый вид животных — «антимышь».

Код ДНК представляется в виде последовательности символов 'A', 'C', 'G' и 'T'. По замыслу учёных, код ДНК «антимыши» должен иметь ту же длину, что и код ДНК мыши, но при этом быть «наименее похожим» на него, то есть наибольшая общая подпоследовательность двух кодов ДНК должна иметь минимальную длину.

При этом символы подпоследовательностей не обязаны в исходной последовательности идти подряд, но обязаны следовать в том же порядке (например, у кодов "САССА" и "СААС" наибольшей общей подпоследовательностью является "САА" или "САС").

По заданному коду ДНК мыши постройте код ДНК «антимыши».

Формат входных данных

В первой строке входного файла задано одно целое число n ($1 \leq n \leq 10^4$) — длина кода ДНК мыши. Во второй строке задан сам код ДНК — строка длины n , состоящая из символов 'A', 'C', 'G', 'T'.

Формат выходных данных

В первой строке выведите одно целое число — требуемую минимальную длину наибольшей общей подпоследовательности. Во второй строке выведите пример кода ДНК «антимыши», для которого соответствующий минимум достигается. В случае наличия нескольких кодов выведите любой из них.

Пример

standard input	standard output
4 GACT	1 AAAA

Задача E. Evaluation

Имя входного файла: standard input
Имя выходного файла: standard output
Ограничение по времени: 15 seconds
Ограничение по памяти: 256 mebibytes

Задано математическое выражение E , в котором присутствуют целочисленные константы от 0 до 9, переменные от a до z , а также операции сложения, умножения и возведения в фиксированную степень (то есть показатель степени должен быть целочисленной константой от 2 до 9), при этом каждая переменная встречается в выражении не более одного раза.

Пусть задано простое число p . Требуется найти количество различных решений уравнения $E = 0$ по модулю p . Иначе говоря, пусть в выражении E задействовано n переменных; из p^n различных вариантов подстановки в выражение для каждой переменной значений $0, 1, \dots, p-1$ выбираются те, для которых вычисленное значение делится на p ; соответственно, требуется подсчитать количество выбранных вариантов. Так как требуемое число может быть слишком велико, выведите остаток от его деления на 30011.

Более формально, определим *выражение* следующим образом:

- Каждое из чисел $0, 1, \dots, 9$ является выражением;
- Каждая из переменных a, b, \dots, z является выражением;
- Если A и B являются выражениями, то $(A+B)$ и $(A*B)$ также являются выражениями. Первое из них обозначает сумму выражений A и B , второе — их произведение.
- Если A — произвольное выражение, а B — число $2, 3, \dots, 9$, то (A^B) является выражением и обозначает A , возведённое в степень B .

Формат входных данных

В первой строке входного файла задано простое число p ($2 \leq p < 15\,000$). Вторая строка содержит выражение E — непустую строку длины не более 300, состоящую из символов '0', '1', ..., '9', 'a', 'b', ..., 'z', '+', '*', '^', '(', ')' и задающую корректное выражение, соответствующее определению из условия.

Формат выходных данных

Выведите одно число — остаток от деления количества различных решений уравнения $E = 0$ по модулю p на 30011.

Пример

standard input	standard output
3 (((a+y)*(z+8))^2)	15

Задача F. Formula-1

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 4 seconds
Ограничение по памяти: 256 mebibytes

В связи с тем, что новые правила «Формулы-1» увеличили количество обгонов, крупнейший интернет-букмекер Бейтландии стала принимать ставки на количество обгонов, совершённых в гонке.

Пусть болиды после квалификации расставлены на стартовом поле в некотором порядке. Тогда ставка является последовательностью чисел a_i , i -е из которых задаёт количество обгонов, совершённых пилотом болида, стартовавшего i -м. При этом считается, что в каждый момент времени пилот может участвовать только в одном обгоне, обгоняя болид, едущий непосредственно впереди него (то есть «тройных» (и более) обгонов не бывает). Обгоны «на круг» за обгоны также не считаются.

После месяца пробной эксплуатации выяснилось, что многие игроки недовольны тем, что система принимает и заведомо невозможные ставки (например, пусть при двух болидах ставка выглядит как 0,2, то есть стартовавший первым должен совершить 0 обгонов, а стартовавший вторым — два, что невозможно принципиально, так как после первого обгона второй уже будет лидировать и обгонять ему будет некого). В связи с этим Вам поступило задание на разработку модуля «защиты от дурака», который бы по введённой ставке определял, не является ли она заведомо невозможной.

Формат входных данных

Входной файл состоит из нескольких тестовых примеров. В первой строке входного файла задано целое число T — количество тестовых примеров. Далее перечислены тестовые примеры, каждый из которых задаётся двумя строками. Первая из них содержит целое число n ($1 \leq n \leq 10^6$) — количество участвующих в гонке болидов, вторая же задаёт ставку, корректность которой Вам предстоит проверить. Во второй строке n целыми числами задаётся ставка. i -е из этих чисел, a_i ($0 \leq a_i \leq 10^9$) определяет количество обгонов, которые должен совершить пилот болида, стартовавшего i -м.

Общий размер входного файла не превышает 20 мегабайт.

Формат выходных данных

Для каждого тестового примера в отдельной строке выведите «ТАК», если заданное в нём распределение обгонов возможно, и «NIE», если ставка является принципиально невозможной.

Пример

standard input	standard output
3	ТАК
2	NIE
0 1	ТАК
3	
0 1 4	
3	
1 1 3	

Задача G. General

Имя входного файла: standard input
Имя выходного файла: standard output
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

После того, как в байтландских газетах прошла информация о том, что археологи недалеко от столицы Байтландии нашли стойбище динозавров, сотни «искателей приключений» поехали в район раскопок, чтобы взять себе на память хотя бы небольшой фрагмент кости динозавра. Ситуация грозила выйти из-под контроля, так что было принято решение о привлечении армии к охране раскопок.

Прибыв на место, генерал, которому была поручена операция, определил некоторое количество *стратегических пунктов* — точек, в которых могут быть размещены часовые. Однако личного состава выделенных для охраны раскопок подразделений не хватало для того, чтобы занять все стратегические пункты. По этой причине некоторые из стратегических пунктов остались незанятыми.

Отчаявшись дождаться подкрепления, генерал решил лично принять участие в охране раскопок в качестве одного из часовых.

Точка на территории раскопок считается охраняемой, если при начале движения из данной точки в произвольном направлении расстояние хотя бы до одного из часовых начинает уменьшаться.

Для того, чтобы выбрать для себя место дислокации, генерал планирует для каждого из незанятых стратегических пунктов узнать площадь охраняемой территории в случае, если генерал займёт этот пункт. Эту часть работы он поручил Вам.

Формат входных данных

В первой строке входного файла заданы два целых числа n и m ($3 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) — количество занятых часовыми стратегических пунктов и количество свободных стратегических пунктов соответственно. В последующих n строках заданы стратегические пункты, занятые часовыми. Каждый пункт задаётся двумя целыми числами — своими координатами x_i и y_i ($-10^8 \leq x_i, y_i \leq 10^8$). Далее в m строках в аналогичном формате заданы свободные стратегические пункты. Гарантируется, что никакие два стратегических пункта с различными номерами не совпадают и что площадь первоначально охраняемой территории является ненулевой.

Формат выходных данных

В выходной файл выведите m строк. В i -й из этих строк выведите площадь охраняемой территории в случае, если генерал займёт соответствующий стратегический пункт. Площадь требуется вывести с точностью до 0.1.

Пример

standard input	standard output
3 2	5.0
0 0	2.5
2 -1	
1 2	
3 1	
1 0	

Задача Н. Hydra

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 3 seconds
Ограничение по памяти: 256 mebibytes

Финальным эпизодом популярной в Байдландии компьютерной игры «12 подвигов рыцаря» является битва рыцаря с гидрой.

В битве рыцарь может выбрать одну из двух тактик: или просто отрубить очередную голову гидры, или «элиминировать» её. В первом случае вместо отрубленной головы отрастают новые, во втором — не вырастают. Головы гидры могут принадлежать к одному из n типов. Каждый тип характеризуется количеством ударов меча для отсечения, количеством ударов меча для элиминации, а также множеством отрастающих при отсечении голов (для каждой из которых определён её тип). Перед началом битвы гидра имеет одну голову первого типа.

Выбрав тактику для некоторой головы, рыцарь не может поменять тактику до того, как голова будет отрублена или элиминирована, равно как и не может переключаться на другие головы. Требуется определить минимальное количество взмахов меча, которое потребуется рыцарю для того, чтобы убить гидру (то есть элиминировать все её головы).

Формат входных данных

В первой строке входного файла задано одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество различных типов головы гидры. Последующие n строк задают соответствующие типы: в i -й строке описывается i -й тип: сначала заданы три целых числа u_i, z_i, h_i — соответственно количество взмахов меча, требуемое для того, чтобы отрубить голову данного типа, количество взмахов меча, требуемое для элиминации головы данного типа и число голов, отрастающих в случае отрубания ($1 \leq u_i < z_i \leq 10^9, 1 \leq r_i$). Далее следуют r_i целых чисел в интервале от 1 до n каждое — типы отрастающих голов.

Гарантируется, что сумма всех r_i не превышает 10^6 .

Формат выходных данных

Выведите одно целое число — минимальное количество взмахов меча, которое потребуется рыцарю для того, чтобы победить гидру.

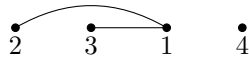
Пример

standard input	standard output
4	26
4 27 3 2 3 2	
3 5 1 2	
1 13 2 4 2	
5 6 1 2	

Задача I. Inversions

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Для заданной перестановки длины n построим содержащий n вершин *граф инверсий* следующим образом. Пусть $V = a_1, a_2, \dots, a_n$ — перестановка (a_i — попарно различные целые положительные числа, не превосходящие n). Вершины с номерами a_i и a_j соединены ребром тогда и только тогда, когда произведение $(a_i - a_j) \cdot (i - j)$ отрицательно. При этом граф является неориентированным.



Требуется перечислить компоненты связности данного графа. Напоминаем, что две вершины u и v принадлежат к одной компоненте связности, если существует такая последовательность вершин, в которой первым элементом является u , последним — v , а любые два соседних элемента соединены ребром.

Формат входных данных

В первой строке входного файла задано одно целое число n ($1 \leq n \leq 10^6$), обозначающее количество вершин в графе. Заданные во второй n целых чисел a_i ($1 \leq a_i \leq n$, $a_i \neq a_j$ для всех $i \neq j$) определяют перестановку.

Формат выходных данных

В первой строке выходного файла выведите одно целое число m — количество компонент связности. В последующих m строках задаются компоненты связности, отсортированные по возрастанию наименьшего номера вершины, входящей в данный компонент. Каждый компонент выводится одной строке, при этом сначала выведите количество вершин в компоненте, а затем список номеров вершин, отсортированный по возрастанию.

Пример

standard input	standard output
4	2
2 3 1 4	3 1 2 3
	1 4

Задача J. Procrastination

Имя входного файла: standard input
Имя выходного файла: standard output
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Проснувшись рано утром после длительных праздников, один программист вспомнил о том, что ему предстоит сделать несколько проектов. Каждый проект характеризуется количеством подряд идущих дней, которые программист должен затратить на его выполнение, а также сроком сдачи — расстоянием от текущего дня до дня сдачи проекта (включительно). При этом «разрывать» время работы над проектом на части или же работать над двумя проектами одновременно он не может.

Так как после особо длинных праздников требуется какое-то время на то, чтобы прийти в себя, программист собирается распланировать работу так, чтобы у него оставалось как можно больше свободных дней перед тем, как он начнёт работать хотя бы над одним из проектов.

Учитывая, что в себя он ещё не пришёл, эту задачу он попросил решить Вас.

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 10^6$) — количество проектов, которые предстоит сделать программисту. В последующих n строках заданы параметры проекта. i -я из этих строк содержит количество дней d_i , затрачиваемых на реализацию i -го проекта, и количество дней t_i , оставшихся до сроков его сдачи ($1 \leq d_i, t_i \leq 10^9$).

Гарантируется, что существует такой план выбора проектов, при котором программист успевает всё сдать в срок.

Формат выходных данных

Выведите одно целое число k — количество дней, по истечении которых программист обязан приступить к работе. Иначе говоря, не позднее, чем в $k + 1$ -й день программист обязан начать работу над некоторым проектом.

Пример

standard input	standard output
3	5
2 8	
1 13	
3 10	

Замечание

В примере из условия на шестой и седьмой день программист начинает работу над первым проектом, на восьмой, девятый и десятый — над третьим, далее в один из оставшихся дней работает над вторым.

Задача K. Rabbits

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

В XIX веке жители Австралии пережили настоящее нашествие кроликов. Расплодившиеся в отсутствие естественных врагов зверьки устраивали набеги на огороды, пожирая всё, что растёт на грядках.

Как-то рано утром фермер Джон проснулся и увидел, что на n грядках, размещённых вокруг его дома, «пасётся» куча кроликов. Грядки на огороде Джона пронумерованы числами от 1 до n , при этом любые две грядки с номерами, отличными на единицу, являются соседними; также соседними являются грядки 1 и n . На грядке с номером i сидели a_i кроликов.

Для борьбы с кроликами Джон может выстрелить из ружья по некоторой грядке. В этом случае кролики с этой грядки убегают в лес, а кролики с каждой из соседних с ней грядок переходят по кругу на одну грядку в сторону «от обстреливаемой».

Учитывая, что у Джона осталось k патронов, вычислите, какое наибольшее количество кроликов он может прогнать в лес.

Формат входных данных

В первой строке входного файла заданы два целых числа n и k ($5 \leq n \leq 2000$, $1 \leq k \leq n$) — количество грядок вокруг дома Джона и количество патронов в его ружье. Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^6$), i -е из которых обозначает количество кроликов, сидящих первоначально на i -й грядке.

Формат выходных данных

Выведите одно число — наибольшее количество кроликов, которое фермер Джон сможет прогнать в лес.

Пример

standard input	standard output
5 2 6 1 5 3 4	13

Замечание

Первым выстрелом Джон прогоняет 6 кроликов с первой грядки. Остальные кролики перераспределяются так: кролик со второй грядки переходит на третью, кролики с пятой — на четвёртую. После чего Джон оставшимся патроном стреляет по четвёртой грядке, прогоняя 7 кроликов