

Задача А. Кто ходит в гости по утрам

Имя входного файла: a.in
Имя выходного файла: a.out
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 Мебибайта

Кто ходит в гости по утрам,
Тот поступает мудро.
Известно всем, тарам-парам,
На то оно и утро! На то оно и утро!

Да, если гость пришёл с утра, –
Ему спешить не надо.
Кричат хозяева “Ура!”,
Они ужасно рады. Они ужасно рады!

Скучна вечерняя пора,
Хозяева зевают.
Но если гость пришёл с утра,
Такого не бывает! Такого не бывает!

Недаром солнце в гости к нам
Всегда приходит по утрам.
Тарам-парам, тарам-парам,
Ходите в гости по утрам.

Б.Заходер, “Песня Винни-Пуха”

В Чудесном Лесу живут N различных персонажей, у каждого из которых есть свой собственный домик. Следуя заветам одного из самых известных лесных персонажей, Винни-Пуха, каждый житель считает необходимым проснуться с утра пораньше, умыться, одеться и пойти в гости к кому-либо. Разумеется, чтобы поступить не просто мудро, а очень мудро и не потратить слишком много времени на дорогу, персонаж отправится не к кому-нибудь, а к своему соседу, то есть к тому из жителей, домик которого находится к данному персонажу на наименьшем возможном расстоянии. Нетрудно понять, что хозяина этого домика не окажется дома, поскольку он тоже воспользуется правилом Винни-Пуха. Лишь по этой причине некому будет ни крикнуть “Ура!”, ни обрадоваться гостям. Если вдруг окажется, что несколько домиков расположены на минимальном расстоянии от персонажа, то он выберет для похода в гости домик с наименьшим номером. Ваша задача – определить какие персонажи соберутся у каждого домика.

Формат входного файла

В первой строке задается количество персонажей N ($2 \leq N \leq 100000$). В каждой из последующих N строк задаются по два числа – координаты точки на плоскости, в которой расположен домик соответствующего персонажа. Все координаты – целые неотрицательные числа, не превосходящие 10^9 .

Формат выходного файла

Выведите N строк. i -ая строка должна содержать число i , за которым следует двоеточие и далее в порядке возрастания номера персонажей, которые придут в гости в i -ый домик.

Пример

a.in	a.out
6	1: 2 3
0 0	2: 1
1 0	3:
0 1	4: 5
3 3	5: 4 6
2 2	6:
3 1	

Задача В. Принц или самозванец

Имя входного файла: `b.in`
Имя выходного файла: `b.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 Мебибайта

В древние времена и минувшие века и столетия был в Персии великий царь Дарий. И процветало в то время персидское государство, и было всего в достатке. В один прекрасный день у царя родился сын, и не было в тот день на земле счастливее человека, чем Дарий. Большой праздник устроил он в честь этого события. Но пока шли все торжества по случаю рождения наследника, нанятые недругами Дария ассасины проникли в опочивальню и выкрали ребенка. Страшно разгневался Дарий и приказал казнить охрану. А кроме того, повелел царь объявить розыск и пообещал огромную награду тому, кто найдет и вернет принца во дворец...

...Шли дни, а за ними недели, месяцы и годы, но никаких вестей о сыне не поступало царю. И вот, когда пошел восемнадцатый год поисков, во дворец вошел стройный высокий юноша с искоркой во взгляде, который назвался тем самым пропавшим принцем. Невероятные истории рассказывал юноша. И о том, как он был подброшен в семью простого рыбака, которому под страхом смерти запретили говорить о принце кому бы то ни было, и о том, как он жил все эти годы, и о том, как, умирая, старый рыбак открыл юноше страшную тайну о его происхождении.

И уже готов был Дарий поверить юноше и принять его в свои объятия, но везирь посоветовал царю не спешить, а сначала сделать проверку. Известно, что каждая клетка организма человека содержит ДНК, которая представляет из себя цепочку нуклеотидов, кодируемых символами **A**, **G**, **T**, **C**, и, кроме того, что эти цепочки у близких родственников должны быть похожими. Везирь предложил взять некоторый фрагмент ДНК юноши и сравнить с ДНК царя, начиная с какой-нибудь позиции. Разумеется, наилучший вариант будет тогда, когда фрагмент в точности встретится начиная с выбранной позиции. Но, вообще говоря, мерой похожести будет считаться количество совпадений при сравнении соответствующих элементов. Везирь с царем просят вас определить позицию, с которой следует начинать сравнение для того, чтобы степень похожести была максимальной.

Формат входного файла

В первой строке задается ДНК царя – последовательность символов **A**, **G**, **T**, **C**. Во второй строке аналогичным образом задается фрагмент ДНК юноши. Длины строк не превышают 200000 и вторая строка не длиннее первой.

Формат выходного файла

Выведите номер позиции, с которой следует начинать сравнение для того, чтобы добиться максимально возможной похожести. Если таких позиций несколько, выведите первую из них. Позиции нумеруются с 1.

Note

Заметьте, что каждому символу второй строки при сравнении должен соответствовать некоторый символ первой строки, иными словами, при размещении второй строки, начиная с нужной позиции, она не выйдет за границы первой.

Примеры

<code>b.in</code>	<code>b.out</code>
AGTCAGTC GTC	2
AAGGTTCC TCAA	5

Задача С. Что? Где? Когда?

Имя входного файла: `c.in`
Имя выходного файла: `c.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мебибайта

“Уважаемые знатоки информатики!” В известной телевизионной игре “Что? Где? Когда?” команда знатоков играет против команды телезрителей, приславших на передачу свои вопросы. Письма с вопросами кладутся на круглый игровой стол, разделенный на N равных секторов – в каждый сектор по одному письму. В центре этого стола установлен волчок со стрелкой. Каждый раунд начинается с того, что распорядитель зала раскручивает волчок. Когда волчок останавливается, из сектора, на котором остановилась стрелка берется конверт и ведущий зачитывает соответствующий вопрос, на который знатоки должны будут ответить после минуты обсуждения. Если же вопрос из сектора, где остановилась стрелка, уже сыграл в одном из предыдущих раундов, то выбирается следующий по часовой стрелке еще не игравший вопрос. Вообще говоря, в телепередаче игра идет до тех пор пока одна из команд не наберет определенного количества очков, но мы будем считать, что игра заканчивается только тогда, когда на столе не останется ни одного вопроса.

Допустим уже прошло несколько раундов и вопросы из некоторых секторов уже сыграли. “А теперь внимание вопрос!” (удар гонга!)

За одну секунду ваша программа должна ответить, какая вероятность того, что в k -ом (начиная с текущего) раунде будет играть вопрос, находящийся в i -ом секторе. Разумеется, поскольку секторы одинаковы, то и остановку стрелки волчка в каждом из них считаем равновероятной.

Формат входного файла

В первой строке задаются три целых числа N, i, k ($1 \leq i \leq N \leq 20, 1 \leq k \leq N$). Во второй строке задаются N чисел, каждое из которых равно либо 0, либо 1. Значение 0 обозначает, что вопрос из соответствующего сектора уже сыграл в одном из предыдущих раундов, 1 – вопрос пока еще на столе.

Формат выходного файла

Выведите вероятность того, что вопрос из i -ого сектора сыграет после k -го вращения волчка с точностью не менее 10^{-8} .

Примеры

<code>c.in</code>	<code>c.out</code>
10 5 1 1 1 1 1 1 1 1 1 1 1	0.1
4 2 2 0 1 1 0	0.25

Задача D. Укладка плит

Имя входного файла: `d.in`
Имя выходного файла: `d.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мебибайта

Известный программист Петя вновь принялся за написание новой компьютерной игры в стиле платформер. На одном из уровней у него есть коридор, разбитый на N равных участков. Этот коридор должен быть покрыт плитами. Одна плита может иметь любую длину и соответственно покрывать несколько последовательных участков. Требуется выполнить укладку плит таким образом, чтобы каждый участок был покрыт заданным числом плит. Помогите Пете сосчитать, какое минимальное количество плит, которое ему понадобится для этого.

Формат входного файла

В первой строке задается целое число N ($1 \leq N \leq 200000$) – длина коридора. Во второй строке записано N целых чисел, каждое из которых определяет количество плит, которыми должен покрываться соответствующий участок. Все числа неотрицательные и не превышают 10^9 .

Формат выходного файла

Выведите минимальное количество плит, которое понадобится для укладки.

Примеры

<code>d.in</code>	<code>d.out</code>
3 3 4 1	4
3 4 1 3	6

Задача E. Прохождение коридора

Имя входного файла: e.in
Имя выходного файла: e.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мебибайта

Как мы уже знаем, в игре Пети есть коридор, разбитый на N участков. Предположим, что каждый из участков покрыт некоторым числом единичных плит. Персонаж в игре, которым управляет игрок, находится в начале коридора перед первым участком и может пройти по этому участку, потратив на это один ход. Если на нем была хотя бы одна плита, то после прохождения по участку одна плита с него исчезает. Таким образом количество плит уменьшится на 1. Если же на участке не было ни одной плиты, то персонаж погибает, соответственно игрок теряет одну жизнь, после чего на этом участке появляется K новых плит, а у игрока появляется новый персонаж в начале коридора. Если игрок удачно прошел участок и не погиб, то он оказывается перед следующим участком, который он может пройти, если на нем есть хотя бы одна плита, или погибнуть, если плит нет. В любом случае потребуется один ход. Разрешается лишь движение вперед. Считается, что игрок прошел коридор, если его персонаж в какой-то момент окажется в конце коридора, то есть пройдет последний участок и не погибнет на нем. Помогите игроку узнать, сколько потребуется жизней и ходов для прохождения коридора.

Формат входного файла

В первой строке даны два целых числа N и K ($1 \leq N \leq 10000$, $1 \leq K \leq 100$) – длина коридора и количество появляющихся после гибели персонажа плит на участке. Во второй строке записано N целых чисел, каждое из которых определяет количество плит, которыми покрыт изначально соответствующий участок. Эти числа могут принимать значения от 0 до K включительно.

Формат выходного файла

Выведите сколько жизней потеряет игрок и сколько ходов он сделает до того момента, когда его персонаж попадет в конец коридора.

Примеры

e.in	e.out
3 3 2 2 2	0 3
4 2 1 0 2 1	2 7
5 1 0 0 0 0 0	31 62

Задача F. Игра

Имя входного файла: `f.in`
Имя выходного файла: `f.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мебибайта

Двое игроков играют в следующую игру. На столе лежит N кучек камней, в i -той кучке в начале n_i камней, кроме того ей приписаны натуральные числа x_i и y_i . Игроки ходят по очереди. За один ход игрок выбирает какую-то кучку. Пусть ее номер i . Тогда он может взять из нее либо x_i , либо y_i камней. Ход может быть выполнен, если кучка содержит не меньше камней, чем игрок собирается из нее взять. Проигрывает тот, кто не может сделать ход. Определите, кто выиграет при правильной игре: игрок, который ходит первым, или игрок, который ходит вторым.

Формат входного файла

В первой строке входного файла задано натуральное число $N \leq 10000$. Следующие N строк содержат по 3 числа каждая. А именно, i -я строка содержит параметры i -той кучки: n_i, x_i, y_i . При этом $1 \leq n_i, x_i, y_i \leq 10^{18}$.

Формат выходного файла

В единственную строку выходного файла выведите "First" (без кавычек), если выиграет первый игрок, и "Second" (без кавычек) иначе.

Примеры

<code>f.in</code>	<code>f.out</code>
4 3 1 2 1 2 3 4 1 1 10 3 5	Second

Задача G. Гамильтонов цикл

Имя входного файла: `g.in`
Имя выходного файла: `g.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 Мебибайта

Однажды граф Уильям Гамильтон решил совершить кругосветное путешествие, в котором он посетил бы N крупнейших городов Земли. Между всеми парами городов есть дороги. Каждой дороге, ведущей от одного города до другого, граф Гамильтон приписывает некоторое число (зрелищность), которое является степенью числа 2. Поскольку по разные стороны от одной дороги пейзажи различны, то зрелищность дороги при проезде по ней в одну сторону отличается от зрелищности при проезде в другую сторону. Более того, оказалось, что все дороги имеют различную зрелищность. Граф хочет выбрать замкнутый маршрут (начинающийся и заканчивающийся в одном и том же городе), проходящий через все города по одному разу и имеющий максимальную суммарную зрелищность. Найдите оптимальный замкнутый маршрут для графа Гамильтона. Граф живет в городе с номером 1.

Формат входного файла

В первой строке задается целое число N – количество городов ($2 \leq N \leq 500$). В каждой из последующих N строк задается по N чисел. Если j -ое число в i -ой строке равно c_{ij} , то зрелищность дороги из города i в город j равна $2^{c_{ij}}$. Все числа c_{ij} ($i \neq j$) различны и находятся в диапазоне от 0 до 10^6 включительно. Значения c_{ii} всегда задаются равными -1 , что означает, что дороги из города i в него же, не проходящей через какой-либо другой город, не существует.

Формат выходного файла

Выведите номера всех городов в той последовательности, в которой графу Гамильтону следует их посетить, чтобы его маршрут имел наибольшую зрелищность. Граф должен начать свое путешествие в городе 1 и закончить в нем же. В случае, если существует несколько оптимальных маршрутов, можно вывести любой из них.

Примеры

<code>g.in</code>	<code>g.out</code>
3 -1 5 1 4 -1 2 6 0 -1	1 2 3 1

Задача Н. Очень дружная группа

Имя входного файла: `h.in`
Имя выходного файла: `h.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мебибайта

В классе N девочек и M мальчиков. Классному руководителю Снежане Денисовне надо выбрать группу из L девочек и K мальчиков на представление класса. Эта группа должна быть очень дружной, то есть каждый выбранный мальчик должен дружить с каждой выбранной девочкой. Снежанну Денисовну интересуют все способы это сделать, а потом она выберет из них наилучший с ее точки зрения. Помогите ей и найдите общее число способов сформировать очень дружную группу детей. Так как ответ может быть очень большим выведите его по модулю 100000007.

Формат входного файла

В первой строке входного файла заданы натуральные числа N, M, L, K . При этом $L \leq N \leq 100000$ и $K \leq M \leq 15$. Следующие N строк содержат по M символов 0 или 1 каждая. При этом j -й символ i -й строки равен 1 тогда только тогда когда i -я девочка дружит с j -м мальчиком.

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

<code>h.in</code>	<code>h.out</code>
4 3 1 1 111 101 110 010	8
4 3 2 1 111 101 110 010	7

Задача I. Деревья с нечетным числом независимых множеств

Имя входного файла: `i.in`
Имя выходного файла: `i.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 Мебибайта

Вам дано натуральное число $n \leq 1000$ и простое число p ($10^7 < p < 10^9$). Найдите количество корневых деревьев из n вершин с непомеченными вершинами, имеющих нечетное число независимых множеств. Результат выведите по модулю p .

Дерево называется **корневым с непомеченными вершинами**, если какая-то его вершина зафиксирована как корень, а порядок сыновей для любой вершины не важен. То есть два дерева считаются равными, если они совпадают после некоторого переупорядочения некорневых вершин. Совсем формальное определение: два корневых дерева с непомеченными вершинами T_1 и T_2 равны, если существует взаимно однозначное отображение f из множества вершин дерева T_1 на множество вершин дерева T_2 , которое переводит корень T_1 в корень T_2 и для любого ребра (u, v) из T_1 в T_2 есть ребро $(f(u), f(v))$.

Некоторое множество вершин графа (возможно пустое) называется **независимым**, если никакие две вершины этого множества не соединены ребром.

Формат входного файла

В единственной строке входного файла заданы числа n и p .

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

	<code>i.in</code>	<code>i.out</code>
1	176531359	0
2	896663687	1
3	793877167	2
51	120107707	114046817

Задача J. Максимальная степень простого

Имя входного файла: `j.in`
Имя выходного файла: `j.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 Мебибайта

Вам дано натуральное число $n > 1$. Рассмотрим все различные простые делители n . Каждый из них входит в разложение n на простые множители в какой-то степени. Требуется найти среди показателей этих степеней максимальный.

Формат входного файла

В первой строке входного файла задано натуральное число $T \leq 500$, количество натуральных чисел n в файле. В последующих T строках заданы сами эти числа. Гарантируется, что каждое из них не превосходит 10^{18} .

Формат выходного файла

Для каждого натурального числа n из входного файла выведите в отдельной строке максимальную степень вхождения простого числа в разложение n на простые множители.

Примеры

<code>j.in</code>	<code>j.out</code>
5	1
2	2
12	3
108	2
36	16
65536	

Задача К. Минимальная степень простого

Имя входного файла: `k.in`
Имя выходного файла: `k.out`
Ограничение по времени: 8 секунд
Ограничение по памяти: 64 Мебибайта

Вам дано натуральное число $n > 1$. Рассмотрим все различные простые делители n . Каждый из них входит в разложение n на простые множители в какой-то степени. Требуется найти среди показателей этих степеней минимальный.

Формат входного файла

В первой строке входного файла задано натуральное число $T \leq 100000$, количество натуральных чисел n в файле. В последующих T строках заданы сами эти числа. Гарантируется, что каждое из них не превосходит 10^{18} .

Формат выходного файла

Для каждого натурального числа n из входного файла выведите в отдельной строке минимальную степень вхождения простого числа в разложение n на простые множители.

Примеры

<code>k.in</code>	<code>k.out</code>
5	1
2	1
12	2
108	2
36	16
65536	