

## Задача А. Равные подстроки

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

Дана строка  $S = s_1 s_2 \dots s_n$  и множество запросов вида  $(l_1, r_1, l_2, r_2)$ . Для каждого запроса требуется ответить, равны ли подстроки  $s_{l_1} \dots s_{r_1}$  и  $s_{l_2} \dots s_{r_2}$ .

Так как вы учитесь в 6 параллели, злые преподаватели требуют, чтобы в решении задачи использовался суффиксный массив и не использовались хэши.

### Формат входных данных

В первой строке записана строка  $S$ , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число  $q$  ( $1 \leq q \leq 100\,000$ ) — количество запросов. В каждой из следующих  $q$  строк записаны числа  $l_1, r_1, l_2, r_2$  ( $1 \leq l_1 \leq r_1 \leq |S|$ ;  $1 \leq l_2 \leq r_2 \leq |S|$ ).

### Формат выходных данных

Для каждого запроса выведите «+», если соответствующие подстроки равны, и «-», в противном случае.

### Примеры

стандартный ввод	стандартный вывод
abacaba 4 1 1 7 7 1 3 5 7 3 4 4 5 1 7 1 7	++-+
qa 3 1 1 1 1 2 2 2 2 1 1 2 2	++-

## Задача В. Персистентная очередь

Имя входного файла: `queue.in`  
Имя выходного файла: `queue.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Реализуйте персистентную очередь.

### Формат входных данных

Первая строка содержит количество действий  $n$  ( $1 \leq n \leq 200\,000$ ). В строке номер  $i + 1$  содержится описание действия  $i$ :

- $1\ t\ m$  — добавить в конец очереди номер  $t$  ( $0 \leq t < i$ ) число  $m$ ;
- $-1\ t$  — удалить из очереди номер  $t$  ( $0 \leq t < i$ ) первый элемент.

В результате действия  $i$ , описанного в строке  $i + 1$  создается очередь номер  $i$ . Изначально имеется пустая очередь с номером ноль.

Все числа во входном файле целые, и помещаются в знаковый 32-битный тип.

### Формат выходных данных

Для каждой операции удаления выведите удаленный элемент на отдельной строке.

### Примеры

<code>queue.in</code>	<code>queue.out</code>
10	1
1 0 1	2
1 1 2	3
1 2 3	1
1 2 4	2
-1 3	4
-1 5	
-1 6	
-1 4	
-1 8	
-1 9	

## Задача С. Кредитные операции

Имя входного файла: `credit.in`  
Имя выходного файла: `credit.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Крупный предприниматель Владимир Дубинин, в недалёком прошлом больше известный как Вован Палёный, контролирует трест из  $N$  предприятий. Бывший поделщик Владимира, а ныне известный банкир Александр Кулаков по прозвищу Саня Кривой владеет холдингом из  $N$  банков. Как и полагается у старых друзей, предприятия г-на Дубинина берут кредиты исключительно у банков г-на Кулакова, в то время как банки г-на Кулакова выдают кредиты только предприятиям г-на Дубинина. Причём с целью уклонения от уплаты налогов вся информация о размерах кредитов тщательно скрывается. Но тут на сцене появляется давний конкурент Владимира и Александра генерал милиции Иван Ломов, когда-то носивший кличку Ваня Гнилой. Г-н Ломов хочет отомстить г-ну Дубинину и г-ну Кулакову за старые обиды и выявить все кредитные операции между их предприятиями и банками.

Для начала люди Ивана произвели выемку документов из офисов предприятий Владимира и для каждого предприятия выяснили суммарный размер кредитов  $SR[i]$ , полученных этим предприятием. Затем в ходе аналогичных операций для каждого банка Александра был установлен суммарный размер кредитов  $SC[j]$ , выданных этим банком. Последним шагом является заполнение так называемой кредитной матрицы на основе полученных данных. В данном случае кредитная матрица представляет собой квадратную таблицу из  $N$  строк и  $N$  столбцов, в которой каждый элемент  $A[i, j]$  должен быть равен размеру кредита, взятого  $i$ -м предприятием г-на Дубинина у  $j$ -го банка г-на Кулакова. Доподлинно известно, что размер любого кредита является целым числом от 0 до 100. Помните, что полученная в ходе следственных действий информация могла быть сфальсифицирована, и тогда заполнить кредитную матрицу не получится.

### Формат входных данных

Первая строка содержит целое число  $N$  ( $2 \leq N \leq 100$ ). Вторая строка содержит  $N$  целых чисел  $SR[i]$  ( $0 \leq SR[i] \leq 32000$ ). Третья строка содержит  $N$  целых чисел  $SC[j]$  ( $0 \leq SC[j] \leq 32000$ ). Сумма всех  $SR[i]$  равняется сумме всех  $SC[j]$ .

### Формат выходных данных

Выведите «NO», если кредитная матрица не может быть заполнена. Иначе в первой строке выведите «YES», а в каждой из следующих  $N$  строк выведите через пробел  $N$  соответствующих элементов  $A[i, j]$  кредитной матрицы. Если задача имеет несколько решений, можно вывести любое из них.

### Примеры

<code>credit.in</code>	<code>credit.out</code>
4	YES
267 157 188 259	100 55 100 12
193 320 346 12	0 70 87 0
	0 95 93 0
	93 100 66 0

## Задача D. Бомбы (простая версия)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Диверсионный отряд выполнил задание и заложил бомбы на территории противника. Теперь нужно найти матожидание количества уничтоженных вражеских юнитов.

Территория противника представляет собой клетчатое поле, на котором расположены вражеские отряды. Каждый отряд выстроен аккуратным прямоугольником со сторонами параллельными осям координат. Прямоугольники не пересекаются между собой. В каждой клетке прямоугольника стоит по одному юниту. Бомбы обладают потрясающей мощностью и могут уничтожить все юниты противника на одной линии с бомбой: вертикальной или горизонтальной, но не обеих. К несчастью, бомбы изготовлены вручную неопытными специалистами, поэтому сказать как точно взорвется каждая бомба (да и взорвется ли) заранее невозможно. Для каждой бомбы известны два числа:  $p_1$  – вероятность горизонтального взрыва,  $p_2$  – вероятность вертикального взрыва. С вероятностью  $100\% - p_1 - p_2$  бомба не взорвется вовсе.

Все бомбы установлены снаружи вражеских отрядов. По данному описанию вражеских отрядов и установленных бомб вам нужно вычислить матожидание числа уничтоженных юнитов.

### Формат входных данных

В первой строке даны два целых числа  $N, M$  – число вражеских отрядов и бомб, соответственно ( $1 \leq N \times M \leq 10^5$ ).

В следующих  $N$  строках содержится описание вражеских отрядов, по одному в строке. Каждый отряд описывается четырьмя целыми числами  $x_1, y_1, x_2, y_2$ : координаты противоположных углов прямоугольника ( $x_1 \leq x_2, y_1 \leq y_2$ ).

В следующих  $M$  строках содержится описание бомб, по одному в строке. Каждая бомба описывается четырьмя целыми числами  $x, y, p_1, p_2$ : координаты бомбы и неотрицательные вероятности взрывов в процентах ( $p_1 + p_2 \leq 100$ ).

Все координаты по модулю не превосходят  $10^9$ .

### Формат выходных данных

В единственной строке выведите ответ с абсолютной или относительной погрешностью не больше  $10^{-9}$ .

## Задача Е. Петя и прямоугольники

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

Маленький Петя очень любит прямоугольники. Петя дал маме список прямоугольников, которые он хочет получить в подарок на Новый Год. Каждый прямоугольник характеризуется  $w$  и высотой  $h$ . Мама хочет сделать Пете приятное и купить все прямоугольники из его списка. Мама отправилась в магазин и узнала, что цена одного прямоугольника равна его площади. К ее счастью, в магазине действует предновогодняя акция, позволяющая покупать прямоугольники не по одному, а сразу наборами. Стоимость одного набора равна ширине самого широкого прямоугольника, умноженной на высоту самого высокого прямоугольника из этого набора. Обратите внимание, что поворачивать прямоугольники (тем самым меняя местами ширину и высоту) нельзя. Помогите маме Пети купить все прямоугольники из списка ее сына, потратив на это наименьшее количество денег.

### Формат входных данных

В первой строке записано число  $N$  ( $1 \leq N \leq 200\,000$ ) — количество прямоугольников в списке Пети. В каждой из следующих  $N$  строк записаны по 2 целых положительных числа, не превышающих  $10^6$  — ширина и высота очередного прямоугольника.

### Формат выходных данных

Выведите одно число — наименьшее количество денег, которое может потратить мама чтобы купить Пете все прямоугольники из его списка.

### Примеры

стандартный ввод	стандартный вывод
4 100 1 15 15 20 5 1 100	500
5 1 10 2 20 3 30 4 40 10 1	170

## Задача F. В бухгалтерии опять всё перепутали

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Лула и Пула пошли получать зарплату. Но в бухгалтерии опять всё перепутали. Лула получил зарплату за Пулу, а Пула ...

Пула не хочет получать за Луну и хочет доказать бухгалтерии, что она не права.

Пула работает в крупной компании «MST Inc.», занимающейся информационным сопровождением «Всеберляндской олимпиады школьников по информатике». В компании «MST Inc.» работает  $n$  сотрудников, причём у каждого из них, кроме самой «MST», есть ровно один непосредственный начальник и несколько (возможно ноль) непосредственных подчинённых.

Всеми начальниками сотрудника компании «MST Inc.» называется множество, состоящее из его непосредственного начальника и множества начальников его непосредственного начальника. Известно, что у каждого сотрудника кроме самой «MST», «MST» входит в множество начальников этого сотрудника.

Множеством подчинённых у сотрудника называется множество, состоящее из него самого и множеств подчинённых у всех непосредственных подчинённых данного сотрудника. В частности, все сотрудники входят в множество подчинённых у «MST».

Каждый месяц каждому сотруднику начисляется зарплата, причём немаленькая, ведь иначе ни один сотрудник не согласился бы работать с «MST». Известно, что в нулевой месяц работы организации, каждому сотруднику заплатили по  $c_i$  бурлей. В качестве поощрения сотрудников «MST» придумала следующее правило: В каждый из следующих  $m$  месяцев берётся сотрудник с номером  $a_i$  и берётся число  $s_i$  — сумма зарплат всех сотрудников во множестве его начальников и подчинённых (включая его самого). Если это число оказывалось слишком большим,  $s_i$  берётся по модулю  $10^9 + 7$ . После этого берётся сотрудник с номером  $b_i$ , и к зарплате всех сотрудников, входящих во множество его начальников и подчинённых (включая его самого) прибавляется число  $s_i$ . С учётом этого изменения платится зарплата в  $i$ -й месяц и пересчитывается зарплата в следующие месяцы.

Вернёмся к Пуле. Пула хочет показать бухгалтерии компании «MST Inc.» что она всё перепутала, а для этого ему надо узнать, сколько же ему должны были заплатить в каждый из месяцев с нулевого по  $m$ -й. К сожалению, в гениальной системе поощрения, разработанной «MST», не может разобраться никто. Поэтому эту задачу поручили вам.

### Формат входных данных

В первой строке входных данных даны 2 числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ) — число сотрудников компании «MST Inc.» и последний день, когда выплачивалась зарплата Пуле.

Во второй строке записано  $n - 1$  число.  $i$ -е из них — номер непосредственного начальника сотрудника номер  $i$  ( $i$  принимает значения от 1 до  $n - 1$ ). При этом «MST» имеет номер 0 и не имеет непосредственного начальника. Пула имеет номер  $n - 1$ .

В третьей строке записано  $n$  чисел  $c_i$  ( $1 \leq c_i \leq 10^9$ ) — зарплата  $i$ -го сотрудника в нулевой день.

В каждой из следующих  $m$  строк записано по 2 числа  $a_i$  и  $b_i$  ( $0 \leq a_i, b_i \leq n - 1$ ) — номер человека, на основе которого происходит поощрение и номер человека, к подчинённым и начальникам которого поощрение применяется (более подробно описано в условии).

### Формат выходных данных

В единственной строке выведите  $m + 1$  число — зарплату Пулы в каждый из дней с 0-го по  $m$ -й. Напоминаем, что Пула имеет номер  $n - 1$ . Обратите внимание, что зарплата **не считается** по модулю  $10^9 + 7$ .

## Примеры

стандартный ввод	стандартный вывод
3 3 0 0 1 1 1 0 0 2 1 1 2	1 4 4 28
4 3 0 1 1 0 1 0 0 0 1 1 3 2 3	0 1 6 20

## Замечание

Пояснение к первому примеру:

В первый день к зарплате каждого сотрудника прибавилось 3 бурля и зарплаты стали соответственно 4, 4, 4.

Во второй день к зарплате сотрудников с номерами 0, 1 прибавилось по 8 бурлей и зарплаты стали соответственно 12, 12, 4.

Во третий день к зарплате сотрудников с номерами 0, 2 прибавилось по 24 бурля и зарплаты стали соответственно 36, 12, 28.

## Задача G. Полные квадраты

Имя входного файла: `perfect-squares.in`  
Имя выходного файла: `perfect-squares.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Множество целых положительных чисел будем называть *полноквадратным*, если произведение его элементов является полным квадратом (равно 1, 4, 9, 16, 25, 36, ...).

Задано множество  $A$ . Определите, сколько непустых подмножеств  $B$  множества  $A$  являются полноквадратными.

### Формат входных данных

В первой строке записано целое число  $N$  — количество элементов множества ( $1 \leq N \leq 100$ ). Во второй строке записаны  $N$  попарно различных чисел  $a_i$  — элементы множества ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно число — количество полноквадратных подмножеств по модулю 1 000 000 007.

### Примеры

<code>perfect-squares.in</code>	<code>perfect-squares.out</code>
4 49 20 500 7	3

## Задача Н. Пасьянс

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Пасьянс или солитер — название группы карточных игр для одного игрока. Один новый безымянный пасьянс играется с колодой карт, на которых написаны целые числа. Игра начинается с перемешивания колоды и раскладывания карт в  $n$  стопок, возможно разной длины.

На каждом ходу игрок может взять верхнюю карту из любой стопки, и положить ее в конец итоговой последовательности карт. После этого в исходной стопке верхней картой становится следующая за убранный, и ее можно использовать на следующем ходу. Игрок никак не может повлиять на итоговую последовательность кроме добавления верхних карт в ее конец, но он знает порядок карт в каждой стопке.

Игра заканчивается, когда игрок выложил все карты из всех стопок в итоговую последовательность. Цель игры — получить лексикографически минимальную итоговую последовательность. Напишите программу, решающую такой пасьянс.

### Формат входных данных

В первой строке записано одно число  $n$  — количество стопок карт ( $1 \leq n \leq 10^3$ ).

В следующих  $n$  строках записано число  $k_i$  — количество карт в  $i$ -й стопке, и  $k_i$  чисел  $a_1, a_2, \dots, a_{k_i}$  — карты в стопке ( $1 \leq k_i \leq 10^3; 0 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите  $\sum_{i=1}^n k_i$  чисел — лексикографически минимальную итоговую последовательность.

### Примеры

стандартный ввод	стандартный вывод
3 1 2 1 100 1 1	1 2 100
2 5 10 20 30 40 50 2 28 27	10 20 28 27 30 40 50
3 2 3 2 3 3 1 1 1 2	2 3 1 1 3 2

## Задача I. ДКА, проверяющий делимость

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В условиях данной задачи разрешим в двоичной записи чисел присутствие ведущих нулей. Кроме того разрешим записывать число 0 в двоичной записи как  $\epsilon$  (как пустую строку).

Таким образом, строки «101» и «000101» являются двоичными записями числа 5, а строки «000000» и « $\epsilon$ » — двоичными записями числа 0.

Дано множество положительных нечетных чисел  $S$ .

Постройте ДКА над алфавитом  $\{0, 1\}$ , принимающий те и только те строки, которые являются двоичными записями чисел, делящихся на хотя бы одно число из множества  $S$ .

Число состояний в построенном автомате не обязано быть минимальным, но не должно превышать 20 000. Гарантируется, что существует искомый ДКА с не более чем 1 000 состояний.

### Формат входных данных

В первой строке содержится число  $|S|$  ( $1 \leq |S| \leq 1000$ ) — мощность множества  $S$ .

Затем следуют  $|S|$  различных положительных нечетных чисел, не превышающих  $10^9$ .

### Формат выходных данных

Выведите описание искомого автомата в следующем формате:

Первая строка содержит число  $|Q|$  — количество состояний автомата ( $1 \leq |Q| \leq 20\,000$ ). (Состояния нумеруются числами от 1 до  $|Q|$ .)

Следующая строка содержит число  $q_0$  ( $1 \leq q_0 \leq |Q|$ ) — номер начального состояния, затем число  $|T|$  — количество терминальных состояний, затем  $|T|$  чисел от 1 до  $|Q|$  — номера терминальных состояний.

Следующие  $|Q|$  строк содержат по  $|\Sigma|$  чисел — описание функции переходов  $\delta$ . Для каждого состояния в отдельной строке приводятся номера состояний, в которые из него ведут переходы по символам 0 и 1.

### Примеры

стандартный ввод	стандартный вывод
3 3 5 15	15 15 7 15 3 5 6 9 10 12 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1
2 2 566	2 2 1 2 2 1 2 1