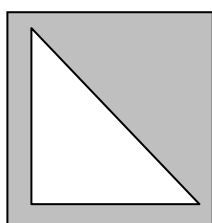


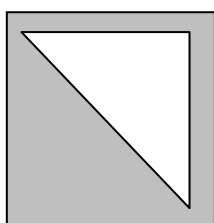
## Задача 1. POBEDA-2014

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

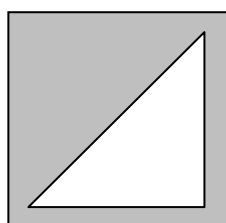
Как известно, современные видеокарты умеют формировать изображения с использованием только треугольников. Видеокарта POBEDA-2014 не отстает от современных тенденций. Известно, что она умеет отображать только прямоугольные равнобедренные треугольники четырех типов ориентации, представленные на рисунках ниже. Изменять ориентацию этих треугольников видеокарта не может.



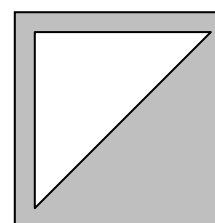
1 тип



2 тип



3 тип



4 тип

Длина катета каждого из представленных выше треугольников равна одному сантиметру. За один такт видеокарта не может отобразить более чем  $a_i$  треугольников  $i$ -того типа.

Необходимо определить максимально возможную длину стороны квадрата, который может быть изображен видеокартой на экране монитора за один такт. При этом квадрат должен быть расположен так, чтобы его стороны были параллельны краям монитора.

Требуется написать программу, которая решает поставленную задачу.

### Формат входного файла

Первая строка входного файла содержит разделенные пробелами четыре целых числа:  $a_1, a_2, a_3, a_4$  ( $0 \leq a_1, a_2, a_3, a_4 \leq 10^{18}$ ). Входные данные могут превышать максимальные значения для 32 битного типа данных.

### Формат выходного файла

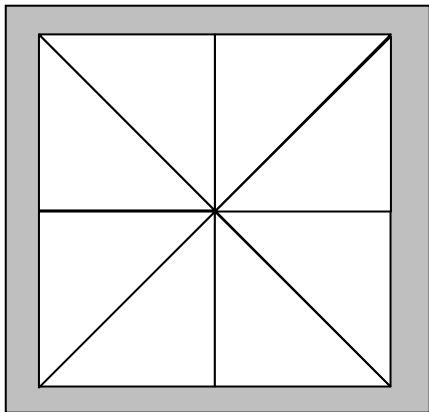
Выходной файл должен содержать одно число – максимально возможную длину стороны квадрата.

### Примеры входных и выходных файлов

<code>pobeda.in</code>	<code>pobeda.out</code>
2 2 2 2	2
10 10 0 0	3

### Пояснения к примерам

Далее приведен рисунок для первого примера.



### **Система оценивания**

Частичные правильные решения для тестов, в которых  $a_1, a_2, a_3, a_4 \leq 100\,000$ , будут оцениваться из 50 баллов.

## Задача 2. Список школ

Имя входного файла: schools.in  
Имя выходного файла: schools.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

При регистрации на портале интернет-олимпиады все участники заполняют регистрационную форму, где они указывают название школы, в которой они учатся. Разные участники могут по-разному писать название школы, например, «Физико-математическая школа №18», «ФМШ №18».

Организаторам олимпиады предоставлена информация о названиях школ, которые написали регистрируемые участники олимпиады. Точно известно, что цифры в названии школы встречаются только в номере школы, а число в записи названия школы встречается ровно один раз и оно однозначно определяет номер школы. Номер школы является положительным целым числом и не может начинаться с нуля.

Требуется написать программу для сайта интернет-олимпиады, которая поможет организаторам олимпиады получить следующую информацию: количество школ и номера школ, из которых зарегистрировалось не более пяти участников.

### Формат входного файла

Первая строка входного файла содержит одно целое число  $n$  ( $1 \leq n \leq 1000$ ) – количество названий школ, указанных всеми участниками при регистрации.

Последующие  $n$  строк содержат названия школ, указанные всеми участниками. Название школы содержит только заглавные и строчные буквы латинского алфавита, цифры и пробелы, длина названия не превышает 100 символов.

### Формат выходного файла

Первая строка выходного файла должна содержать одно число  $m$  – количество школ, от которых на олимпиаду зарегистрировалось от одного до пяти участников. Последующие  $m$  строк должны содержать только номера таких школ, при этом номера должны располагаться по одному в строке в произвольном порядке.

### Примеры входных и выходных файлов

<code>schools.in</code>	<code>schools.out</code>
9	2
Physics and Mathematics School 18	42
9ya shkola imeni Pushkina	18
Lyceum 9	
PaMS 18	
Gymnasium 42	
School 9	
Shkola nomer 9	
High school 9	
School N 9	

### Пояснения к примерам

В приведенном примере для участия в интернет-олимпиаде зарегистрировались: два ученика из школы с номером 18, один ученик из школы с номером 42 и шесть учеников из школы с номером 9. Таким образом, от 1 до 5 участников зарегистрировано от школ с номерами 18 и 42.

### **Система оценивания**

Частичные правильные решения для тестов, в которых все номера школ являются однозначными числами, будут оцениваться из 30 баллов.

Частичные правильные решения для тестов, в которых номера школ – это числа строго меньше 1000, будут оцениваться из 50 баллов.

Частичные правильные решения для тестов, в которых номера школ – это числа строго меньше  $10^9$ , будут оцениваться из 80 баллов.

Несмотря на выделение отдельных групп тестов, на окончательную проверку будут приниматься только решения, правильно работающие для теста из условия задачи.

### Задача 3. Межрегиональная олимпиада

Имя входного файла:	olympiad.in
Имя выходного файла:	olympiad.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На межрегиональной олимпиаде по программированию роботов соревнования проводятся в один тур и в необычном формате. Задачи участникам раздаются последовательно, а не все в самом начале тура, и каждая  $i$ -я задача ( $1 \leq i \leq n$ ) становится доступной участникам в свой момент времени  $s_i$ . При поступлении очередной задачи каждый участник должен сразу определить, будет он ее решать или нет. В случае, если он выбирает для решения эту задачу, то у него есть  $t_i$  минут на то, чтобы сдать ее решение на проверку, причем в течение этого времени он не может переключиться на решение другой задачи. Если же участник отказывается от решения этой задачи, то в будущем он не может к ней вернуться. В тот момент, когда закончилось время, отведенное на задачу, которую решает участник, он может начать решать другую задачу, ставшую доступной в этот же момент, если такая задача есть, или ждать появления другой задачи. При этом за правильное решение  $i$ -й задачи участник получает  $c_i$  баллов.

Артур, представляющий на межрегиональной олимпиаде один из региональных центров искусственного интеллекта, понимает, что важную роль на такой олимпиаде играет не только умение решать задачи, но и правильный стратегический расчет того, какие задачи надо решать, а какие пропустить. Ему, как и всем участникам, до начала тура известно, в какой момент времени каждая задача станет доступной, сколько времени будет отведено на ее решение и сколько баллов можно получить за ее решение. Артур является талантливым школьником и поэтому сможет успешно решить за отведенное время и сдать на проверку любую задачу, которую он выберет для решения на олимпиаде.

Требуется написать программу, которая определяет, какое максимальное количество баллов Артур сможет получить при оптимальном выборе задач, которые он будет решать, а также количество и перечень таких задач.

#### **Формат входного файла**

Первая строка входного файла содержит одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) - количество задач на олимпиаде.

Последующие  $n$  строк содержат описания задач, по три числа на каждой строке:  $s_i$  - момент появления  $i$ -й задачи в минутах,  $t_i$  - время, отведенное на ее решение в минутах, и  $c_i$  - сколько баллов получит участник за решение этой задачи ( $1 \leq s_i, t_i, c_i \leq 10^9$ ).

#### **Формат выходного файла**

Первая строка выходного файла должна содержать одно число - максимальное количество баллов, которое сможет получить Артур на олимпиаде.

Вторая строка должна содержать одно целое число  $m$  - количество задач, которые надо решить при оптимальном выборе.

Третья строка должна содержать  $m$  разделенных пробелом целых чисел - номера этих задач в порядке их решения. Задачи пронумерованы, начиная с единицы, в порядке их описания во входном файле.

Если оптимальных ответов несколько, необходимо вывести любой из них.

### **Примеры входных и выходных файлов**

<code>olympiad.in</code>	<code>olympiad.out</code>
2 1 1 1 2 2 2	3 2 1 2
3 1 2 1 3 2 1 2 4 3	3 1 3

### **Пояснения к примерам**

В первом примере Артур успеваеt решить все задачи и получить три балла.

Во втором примере Артуру выгоднее решать последнюю задачу и получить за нее три балла, чем решать только первые две и получить два балла.

### **Система оценивания**

Частичные правильные решения для тестов, в которых все  $c_i$  одинаковы и  $n \leq 1000$ , оцениваются из 30 баллов.

Частичные правильные решения для тестов, в которых все  $c_i$  одинаковы, оцениваются из 50 баллов.

Частичные правильные решения для тестов, в которых  $n \leq 1000$ , оцениваются из 50 баллов.

## Задача 4. Дом Мэра

Имя входного файла:	majorhouse.in
Имя выходного файла:	majorhouse.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

При планировании нового района города М было решено, что дороги в новом районе будут образовывать прямоугольную сетку, то есть все улицы будут одного из двух типов – направленные с юга на север или направленные с востока на запад. При этом параллельные улицы будут проходить через каждый километр, и каждый квартал будет иметь размер ровно один километр на один километр. Таким образом, все дороги образуют равномерную клетчатую сетку. По каждой дороге разрешен проезд в любом из двух направлений.

Через некоторое время после постройки дорог оказалось, что такая планировка не всегда удобна, поскольку для постройки больших заводов или других сооружений и организации парков недостаточно одного квартала. Мэрия города М решила отдать каждому большому проекту по прямоугольному блоку из нескольких соседних кварталов. К сожалению, после реализации проекта все дороги внутри такого блока будут закрыты для проезда, но по границе блоков проезд все еще будет возможен. Касание двух блоков не закрывает проезд между ними.

Когда Мэру города М принесли на согласование план распределения территорий для больших проектов, ему стало интересно, насколько сложным будет маршрут от мэрии до его будущего дома. Мэрия находится в центре нового района, на пересечении нулевой улицы, направленной с юга на север, и нулевой улицы, направленной с востока на запад. С итоговым расположением дома Мэр еще не определился и на выбор у него есть  $k$  вариантов. Каждый из вариантов находится на пересечении  $x_i$ -ой улицы, направленной с юга на север (положительный  $x$  означает, что улица находится восточнее мэрии, отрицательный – западнее) и  $y_i$ -ой улицы, направленной с востока на запад (положительный  $y$  означает, что улица находится севернее мэрии, отрицательный – южнее).

Мэр считает, что маршрут до дома является сложным, если ему на этом маршруте придется совершить более двух поворотов направо или налево. Машина Мэра не может совершать более одного поворота на перекрестке, например, чтобы развернуться. Длина маршрута не имеет значения, и к дому можно подъезжать с любой стороны. Машина Мэра всегда стоит у мэрии в северном направлении, может повернуть сразу направо или налево, но не может развернуться.

Требуется написать программу, которая по информации о закрытых для проезда блоках кварталов и возможным расположениям дома Мэра, для каждого возможного расположения дома Мэра найдет несложный маршрут от мэрии до дома, определит кратчайший из них, или сообщит, что такого маршрута не существует. Количество поворотов минимизировать не требуется.

### Формат входного файла

Первая строка входного файла содержит два целых числа  $n$  и  $k$  ( $0 \leq n \leq 100\,000$ ,  $1 \leq k \leq 10$ ) – количество блоков кварталов, которые по плану будут отданы большим проектам и количество вариантов расположения дома Мэра, соответственно.

Последующие  $n$  строк содержат по описанию блоков кварталов – четыре целых числа  $u_1, v_1, u_2, v_2$  ( $-10^9 \leq u_1 < u_2 \leq 10^9$ ,  $-10^9 \leq v_1 < v_2 \leq 10^9$ ) — номера улиц, на пересечении которых расположены противоположные углы блока кварталов, отданных под застройку и закрытых для проезда.

Последующие последние  $k$  строк содержат по два целых числа  $x_i$  и  $y_i$  ( $|x_i| \leq 10^9$ ,  $|y_i| \leq 10^9$ ,  $x_i \neq 0$  или  $y_i \neq 0$ ) — возможные расположения дома Мэра.

Мэрия и никакое из возможных расположений дома Мэра не находятся внутри блоков кварталов, отданных под застройку, но блоки кварталов, отданные под застройку, могут пересекаться.

### Формат выходного файла

В выходной файл для каждого из возможных расположений дома Мэра в порядке появления во входном файле необходимо вывести сообщение, существуют ли несложный маршрут от мэрии до дома Мэра и, если существует, то где надо сделать повороты.

Если не существует несложный маршрут, то сообщение должно содержать слово NO на одной строке. Иначе, в первой строке должно содержаться слово YES, во второй строке — одно число  $t$  ( $0 \leq t \leq 2$ ) - количество поворотов, и в последующих  $t$  строках — описания поворотов в порядке их совершения: в каждой строке по три числа  $x$ ,  $y$  и  $d$  - номера улиц, на которых расположен перекресток, где необходимо повернуть, и направление поворота, при этом  $d = -1$  означает поворот налево и  $d = 1$  — поворот направо.

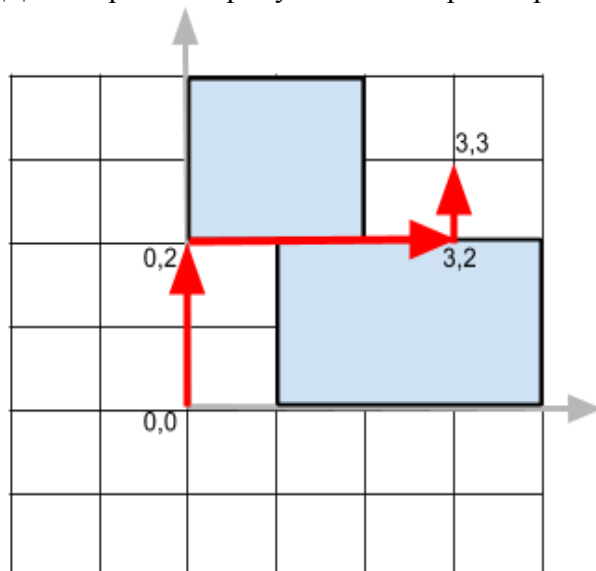
Координаты перекрестков, где необходимо совершить повороты, не должны превышать  $10^9$ . Если кратчайших несложных путей несколько, необходимо вывести любой из них.

### Примеры входных и выходных файлов

majorhouse.in	majorhouse.out
1 2 -2 1 9 2 2 0 3 3	YES 1 0 0 1 NO
2 1 0 2 2 4 1 0 4 2 3 3	YES 2 0 2 1 3 2 -1
0 2 0 -1 0 1	NO YES 0

### Пояснения к примерам

Далее приведен рисунок для второго примера.





### **Система оценивания**

Частичные правильные решения для тестов, в которых все координаты ( $x$ ,  $y$ ,  $u$  и  $v$ ) по модулю не превышают 100, и  $n \leq 50$ , будут оцениваться из 30 баллов.

Частичные правильные решения для тестов, в которых  $n \leq 50$ , будут оцениваться из 60 баллов.

## Задача 5. Светофоры

Имя входного файла: lights.in  
Имя выходного файла: lights.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

По территории компьютерного лагеря проложен маршрут для поездок на электрокарах. Поскольку на электрокаре можно добраться до ИКТ-центра, то школьник Пахом решил воспользоваться им. Следуя по маршруту, электрокар проехал с постоянной скоростью один за другим два светофора с зеленым светом. Пахому известно, что оба светофора находятся на расстоянии  $x$  метров друг от друга и переключаются абсолютно синхронно: зеленый свет горит  $a$  минут, потом включается красный свет и горит в течение  $b$  минут, после чего светофор переключается опять на зеленый свет и он горит также в течение  $a$  минут, и так далее. Переключений на желтый свет у светофоров нет. Скорость движения электрокара по маршруту не превышает 1000 м/мин. Электрокар может проехать на светофоре в тот момент, когда светофор переключается с одного света на другой.

Приехав в ИКТ-центр, Пахом заинтересовался, с какой максимальной постоянной скоростью он мог ехать на электрокаре между двумя светофорами.

Требуется написать программу, которая позволит Пахому выяснить это.

### Формат входного файла

Первая строка входного файла содержит три целых числа:  $a$ ,  $b$  и  $x$  ( $1 \leq a \leq 100$ ,  $1 \leq b \leq 100$ ,  $1 \leq x \leq 100\,000$ ).

### Формат выходного файла

Выходной файл должен содержать одно число – максимальную возможную скорость электрокара между двумя светофорами. Ответ должен отличаться от правильного не более чем на  $10^{-9}$ .

### Примеры входных и выходных файлов

lights.in	lights.out
3 5 4000	800
5 10 21010	840.4

### Система оценивания

Правильные решения для тестов, в которых ответ является целочисленным, будут оцениваться из 50 баллов.

Несмотря на выделение отдельной группы тестов для целочисленных ответов, на окончательную проверку будут приниматься только решения, правильно работающие для всех тестов из условия задачи.

## Задача 6. Кондиционеры

Имя входного файла: cond.in  
Имя выходного файла: cond.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

При реализации проекта «Умная школа» было решено в каждый учебный класс выбранной для этого школы установить по кондиционеру нового поколения для автоматического охлаждения и вентиляции воздуха. По проекту в каждом классе должен быть установлен только один кондиционер и мощность кондиционера должна быть достаточной для размеров класса. Чем больше класс, тем мощнее должен быть кондиционер.

Все классы школы пронумерованы последовательно от 1 до  $n$ . Известно, что для каждого класса с номером  $i$ , требуется ровно один кондиционер, мощность которого больше или равна  $a_i$  ватт.

Администрации школы предоставили список из  $m$  различных моделей кондиционеров, которые можно закупить. Для каждой модели кондиционера известна его мощность и стоимость. Требуется написать программу, которая определит, за какую минимальную суммарную стоимость кондиционеров можно оснастить все классы школы.

### Формат входного файла

Первая строка входного файла содержит одно целое число  $n$  ( $1 \leq n \leq 50\,000$ ) - количество классов в школе.

Вторая строка содержит  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 1000$ ) - минимальная мощность кондиционера в ваттах, который можно установить в классе с номером  $i$ .

Третья строка содержит одно целое число  $m$  ( $1 \leq m \leq 50\,000$ ) - количество предложенных моделей кондиционеров.

Далее, в каждой из  $m$  строк содержится пара целых чисел  $b_j$  и  $c_j$  ( $1 \leq b_j \leq 1000$ ,  $1 \leq c_j \leq 1000$ ) - мощность в ваттах  $j$ -й модели кондиционера и его цена в рублях соответственно.

### Формат выходного файла

Выходной файл должен содержать одно число - минимальную суммарную стоимость кондиционеров в рублях. Гарантируется, что хотя бы один корректный выбор кондиционеров существует, и во всех классах можно установить подходящий кондиционер.

### Примеры входных и выходных файлов

cond.in	cond.out
1 800 1 800 1000	1000
3 1 2 3 4 1 10 1 5 10 7 2 3	13

### ***Пояснения к примерам***

В первом примере нужно купить один единственно возможный кондиционер за 1000 рублей.

Во втором примере оптимально будет установить в первом и втором классах кондиционеры четвертого типа, а в третьем классе – кондиционер третьего типа. Суммарная стоимость этих кондиционеров будет составлять 13 рублей (3 + 3 + 7).

### ***Система оценивания***

Частичные решения для  $n, m \leq 1000$  будут оцениваться из 50 баллов.

## Задача 7. Конфеты

Имя входного файла: sweets.in  
Имя выходного файла: sweets.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Кондитерская фабрика города П, в котором живет Петя, делает очень вкусные конфеты. Как-то раз, Петя собрался в гости к своему другу Васе, который живет в городе М. От города П до города М Петя решил доехать на поезде и взять с собой в подарок как можно больше коробок вкусных конфет.

Каждая коробка конфет имеет размеры  $a \times b \times c$  сантиметров, где  $a$  – длина,  $b$  – ширина и  $c$  – высота коробки. Для перевозки конфет Петя хочет использовать один большой ящик в форме прямоугольного параллелепипеда. В ящик должны быть уложены все коробки конфет. Для того чтобы не повредить их, все коробки в ящике должны сохранять исходную ориентацию и располагаться в одном направлении. Петя может использовать ящик любого размера, но по правилам железнодорожных перевозок размер ящика по сумме трех измерений не может превышать  $N$  сантиметров.

Требуется написать программу, которая по заданным числам  $N$ ,  $a$ ,  $b$  и  $c$  определяет такой размер ящика, который должен использовать Петя, чтобы в него поместилось максимальное количество коробок конфет.

### Формат входного файла

Первая строка входного файла содержит разделенные пробелами четыре целых числа:  $N$ ,  $a$ ,  $b$ ,  $c$  ( $1 \leq N, a, b, c \leq 10^9$ ).

### Формат выходного файла

Выходной файл должен содержать три числа – длину, ширину и высоту ящика, который должен выбрать Петя и в который поместится максимальное количество коробок конфет. Если подходящих ответов несколько, необходимо вывести любой.

### Примеры входных и выходных файлов

sweets.in	sweets.out
10 1 2 3	3 4 3
14 8 3 2	9 3 2

### Пояснения к примерам

В первом примере выгоднее всего взять ящик размером  $3 \times 4 \times 3$  сантиметров, в который поместится три коробки конфет в длину, две коробки конфет в ширину и одна коробка конфет в высоту.

Во втором примере для того, чтобы разместить хотя бы две коробки конфет, нужен ящик размером хотя бы  $8 \times 3 \times 4$ , у которого сумма измерений равна 15. В подходящий ящик поместится максимум одна коробка конфет. Подходящим также является ящик размером  $9 \times 3 \times 2$ , хотя он и не является минимальным.

### Система оценивания

Частичные правильные решения для тестов, в которых  $N \leq 1000$ , будут оцениваться из 30 баллов.

Частичные правильные решения для тестов, в которых  $N \leq 100\,000$ , будут оцениваться из 60 баллов.

## Задача 8. Волонтеры

Имя входного файла:	volunteers.in
Имя выходного файла:	volunteers.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В проведении межпланетной олимпиады по информатике принимает участие большое количество андроидов. Андроиды, придумывающие задачи, пишущие условия и разрабатывающие наборы тестов, являются членами научного комитета. Те андроиды, которые обеспечивают работоспособность компьютеров и установленного на них программного обеспечения, являются членами технического комитета. Есть третья большая группа андроидов, участвующих в организации и проведении многих других мероприятий, – это волонтеры.

У каждого волонтера есть свой непосредственный начальник в научном комитете и свой непосредственный начальник в техническом комитете. У каждого члена научного комитета, кроме его председателя, есть свой непосредственный начальник – другой член научного комитета. У каждого члена технического комитета, кроме его председателя, также есть свой непосредственный начальник, являющийся членом технического комитета.

В самом начале олимпиады происходит процесс, определяющий начальников для всех волонтеров, членов научного комитета и членов технического комитета. Происходит он следующим образом.

Все  $n$  волонтеров получают уникальные номера от одного до  $n$ , после чего выстраиваются друг за другом в порядке возрастания своих номеров. После этого  $m$  членов научного комитета, также имеющие уникальные номера от одного до  $m$ , по очереди, в порядке возрастания своих номеров, выбирают себе подчиненных следующим образом. Вначале, первый член научного комитета выбирает себе в подчинение некоторое количество волонтеров, стоящих в данный момент в строю подряд друг за другом. После этого те, кого он выбрал, выходят из строя, а он сам встает в строй на то место, где стояли его новоиспеченные подчиненные, заменяя собой сразу несколько андроидов в строю. Затем второй член научного комитета выбирает себе подчиненных из тех, кто теперь стоит в строю и заменяет их собой аналогично первому. Так этот процесс продолжается и каждый последующий член научного комитета может выбрать себе в подчинение находящихся в строю волонтеров и членов научного комитета. Последним выбирает себе подчиненных председатель научного комитета, и он должен выбрать всех андроидов, кто на тот момент будет находиться в строю.

После того, как в строю останется только председатель научного комитета, аналогичный процесс повторяется с самого начала, но уже с членами технического комитета. Важно то, что номера волонтеров при этом сохраняются и волонтеры выстраиваются в том же порядке, что и при выборе подчиненных членами научного комитета.

Считается, что член научного комитета может конфликтовать с членом технического комитета, если один и тот же волонтер подчиняется, возможно, не напрямую, и члену научного комитета, и члену технического комитета. Из этого определения, в частности, следует, что председатель научного комитета всегда может конфликтовать с председателем технического комитета. Несложно понять, что от общего количества таких возможных конфликтов между андроидами зависит то, насколько качественно будет проведена олимпиада.

Требуется написать программу, которая для каждого члена научного комитета подсчитывает количество членов технического комитета, с которыми он может конфликтовать, и определяет суммарное количество таких конфликтов.

### **Формат входного файла**

Первая строка входного файла содержит числа  $n$ ,  $m$  и  $k$  ( $1 \leq n, m, k \leq 100\,000$ ) – количество волонтеров, членов научного комитета и членов технического комитета соответственно.

Последующие  $n$  строк содержат по два числа  $a_i$  и  $b_i$  ( $1 \leq a_i \leq m, 1 \leq b_i \leq k$ ) – номер начальника соответствующего волонтера среди членов научного комитета и номер начальника соответствующего волонтера среди членов технического комитета.

Следующая строка содержит  $(m-1)$  чисел  $c_i$  ( $i = 0 \dots (m-1), i < c_i \leq m$ ) – номер начальника  $i$ -го члена научного комитета. Председатель научного комитета имеет номер  $m$ .

Следующая строка содержит  $(k-1)$  чисел  $d_i$  ( $i = 0 \dots (k-1), i < d_i \leq k$ ) – номер начальника  $i$ -го члена технического комитета. Председатель технического комитета имеет номер  $k$ .

Гарантируется, что все входные данные корректны, то есть они получены в результате процесса назначения подчиненных, описанного выше.

### **Формат выходного файла**

Первая строка выходного файла должна содержать одно число – суммарное количество искоемых конфликтов.

### **Примеры входных и выходных файлов**

<code>volunteers.in</code>	<code>volunteers.out</code>
4 3 4 1 2 1 1 2 1 2 4 3 3 3 3 4	11

### **Пояснения к примерам**

В представленном примере только второй член научного комитета и второй член технического комитета не могут конфликтовать друг с другом.

### **Система оценивания**

Частичные правильные решения для тестов, в которых количества волонтеров, членов научного комитета и членов технического комитета не превышают 100, будут оцениваться из 30 баллов.

Частичные правильные решения для тестов, в которых количества волонтеров, членов научного комитета и членов технического комитета не превышают 5000, будут оцениваться из 60 баллов.