

## Задача А. Профессор Хаос

Имя входного файла: `chaos.in`  
Имя выходного файла: `chaos.out`

В секретной лаборатории профессора Хаоса проходит эксперимент по выращиванию особо опасных бактерий. В начале первого дня эксперимента у Хаоса имеется  $a$  особо опасных бактерий.

Каждый день эксперимента устроен следующим образом. Рано утром профессор достает из контейнера все свои бактерии и помещает их в инкубатор, где бактерии начинают делиться. Вместо каждой бактерии образуется  $b$  новых бактерий.

После извлечения бактерий из инкубатора  $c$  из них используются для проведения различных опытов и затем уничтожаются. Если после извлечения из инкубатора имеется менее  $c$  бактерий, для проведения опытов используются все имеющиеся бактерии, и эксперимент заканчивается.

Оставшиеся бактерии в конце дня необходимо поместить в контейнер и продолжить использовать в эксперименте. Однако в контейнер можно поместить не более  $d$  бактерий, поэтому если число оставшихся бактерий больше  $d$ , то в контейнер помещаются  $d$  бактерий, а остальные уничтожаются.

Теперь профессор Хаос хочет выяснить, сколько особо опасных бактерий будет у него в контейнере после  $k$ -го дня эксперимента. Помогите ему найти ответ на этот вопрос.

### Формат входных данных

В единственной строке входного файла содержится пять целых чисел  $a, b, c, d$  и  $k$  ( $1 \leq a, b \leq 1000$ ,  $0 \leq c \leq 1000$ ,  $1 \leq d \leq 1000$ ,  $a \leq d$ ,  $1 \leq k \leq 10^{18}$ ).

### Формат выходных данных

Выведите одно число — количество бактерий у Хаоса к концу  $k$ -го дня. Если эксперимент завершится в  $k$ -й день или ранее, выведите число 0.

### Примеры

<code>chaos.in</code>	<code>chaos.out</code>
1 3 1 5 2	5
1 2 0 4 3	4
1 2 3 5 2	0

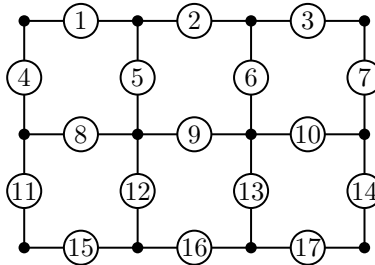
## Задача В. Золотые монеты

Имя входного файла: `coins.in`  
Имя выходного файла: `coins.out`

Саша очень любит играть в игры на своем планшете. Недавно он скачал новую игру, которая называется «Золотые монеты».

Игра проходит на поле, имеющем вид сетки с тремя горизонтальными и четырьмя вертикальными рядами. Это поле представляет собой город: линии сетки являются дорогами, а узлы сетки — перекрестками. По дорогам можно перемещаться в обоих направлениях. В начале игры на каждой дороге расположено несколько золотых монет.

Пример игрового поля приведен на рисунке, перекрестки обозначены черными точками, дороги — отрезками, а число на дороге задает начальное количество золотых монет на ней.



Персонажем игры является бородатый электромонтер Томас. Сначала игрок может поместить Томаса на любой перекресток. Затем каждый ход игрок перемещает Томаса с перекрестка, на котором он находится, на соседний перекресток по одной из дорог, на которой еще лежит хотя бы одна монета.

Проходя по дороге, Томас забирает округленную вверх половину лежащих на ней монет. Таким образом, если на дороге лежит  $x$  монет, то, пройдя по этой дороге, Томас заберет  $\lceil x/2 \rceil$  монет, а на дороге останется  $\lfloor x/2 \rfloor$  монет. По дорогам, на которых монет уже нет, перемещать Томаса не разрешается.

Когда Томас оказывается в ситуации, что на всех соседних дорогах не осталось ни одной монеты, игра заканчивается. Очки игрока равны числу монет, которые собрал Томас.

Помогите Саше понять, какое максимальное количество монет он сможет собрать.

### Формат входных данных

Входной файл состоит из пяти строк. Первая, третья и пятая строки содержат по три целых числа и описывают соответствующие горизонтальные улицы. Вторая и четвертая строки содержат по четыре целых числа и описывают соответствующие вертикальные улицы. Все числа неотрицательные и не превосходят  $10^9$ .

### Формат выходных данных

В единственной строке выходного файла выведите число  $s$  — максимальное количество монет, которые удастся собрать.

### Примеры

<code>coins.in</code>	<code>coins.out</code>
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	150
1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 1 1	7

## Задача С. Вещественные числа

Имя входного файла: `fraction.in`  
Имя выходного файла: `fraction.out`

Первое правило вещественных чисел — не используйте вещественные числа

Подслушано в Заколково

Петя работает в наукограде Заколково. Он занимается разработкой нового микропроцессора Чебур. Поскольку ведущий архитектор микропроцессора очень не любит числа с плавающей точкой, вещественные числа в микропроцессоре Чебур хранятся в формате с фиксированной точкой.

Внутренние регистры процессора хранят вещественные числа в виде двоичной дроби ровно с  $n$  двоичными знаками после точки. Число знаков до точки не ограничено. В таком формате представимы числа, равные  $p/2^n$  для некоторого целого  $p$ . Число  $x$ , полученное в результате выполнения арифметических операций над вещественными числами, при сохранении в регистр процессора заменяется на ближайшее представимое число, а если  $x$  находится ровно посередине между двумя представимыми числами, то на большее из них.

Для хранения чисел в памяти используется формат, при котором вещественное число представляется в виде двоичной дроби ровно с  $k$  двоичными знаками после точки ( $k \leq n$ , число знаков до точки, так же как и у регистров, не ограничено). В таком формате представимы в точности числа, равные  $p/2^k$  для некоторого целого  $p$ . При загрузке числа из памяти в регистр процессора число дополняется нулями до  $n$  двоичных знаков после точки. При сохранении числа  $y$  в память из регистра оно заменяется на ближайшее представимое число, а если  $y$  находится ровно посередине между двумя представимыми числами, то на большее их них.

Например, пусть  $n = 10$ ,  $k = 5$ . Число 1 хранится в памяти в виде  $1.00000_2$ , число 3 хранится в памяти как  $11.00000_2$ . При загрузке в регистры числа дополняются нулями до  $1.0000000000_2$  и  $11.0000000000_2$ , соответственно. Пусть было выполнено деление 1 на 3. Если результат деления  $1/3$  записать в двоичной системе, получится  $0.(01)_2$  — здесь, как и в случае десятичных дробей, часть в скобках означает период бесконечной двоичной дроби. При сохранении в регистре процессора эта дробь будет приближена значением  $0.01010101_2$ . Умножим теперь это число на 3. Получится число  $0.11111111_2$ , в таком же виде оно хранится в регистре процессора. При сохранении в память оно приближается числом  $1.00000_2$ , как ближайшей двоичной дробью с 5 знаками после точки.

Петя заметил, что если загрузить в регистры единицу и целое число  $v$ , поделить 1 на  $v$ , затем умножить результат деления на  $v$  и сохранить результат умножения в память, то итоговое сохраненное значение не всегда равно 1. Например, если  $v = 40$ , то после деления в регистре оказывается значение  $0.0000011010_2$ , после умножения на  $v$  значение  $1.0000010000_2$ , после сохранения в память оно преобразуется в  $1.00001_2$ . Петя называет такие числа *неудачными*.

Петю заинтересовал вопрос, какие целые числа от 1 до  $r$  являются неудачными. Помогите ему выяснить это.

### Формат входных данных

Первая строка входного файла содержит три целых числа  $n$ ,  $k$  и  $r$  ( $1 \leq k \leq n \leq 100$ ,  $1 \leq r \leq 1000$ ).

### Формат выходных данных

В первой строке выходного файла выведите число  $t$  — количество неудачных чисел, лежащих в диапазоне от 1 до  $r$ . Во второй строке выходного файла через пробел выведите в возрастающем порядке все неудачные числа, лежащие в диапазоне от 1 до  $r$ .

### Примеры

<code>fraction.in</code>	<code>fraction.out</code>
10 5 60	7 40 50 52 53 55 58 59

## Задача D. Игра

Имя входного файла: `game.in`  
Имя выходного файла: `game.out`

Пете подарили на день рождения новую карточную игру «Салют». Эта игра предназначена для одного человека. В комплект игры входит колода, состоящая из  $n$  карт. На каждой карте написано целое число от 1 до  $m$ .

Игра происходит следующим образом. Колода карт перемешивается, и игрок берет в руку верхние  $k$  карт из колоды. В каждый момент времени игрок может держать в руке не более  $k$  карт. Есть три различных вида хода.

- Сбросить любую карту, находящуюся в руке. Сброшенная карта отправляется в снос и не может быть далее использована в игре.
- Если в колоде еще есть карты, то взять в руку верхнюю карту колоды. Такой ход можно сделать только, если в руке у игрока строго меньше, чем  $k$  карт.
- Выложить карту из руки на стол. Карту с числом  $x$  можно выложить в том случае, если игрок до этого уже выложил на стол карты с числами  $1, 2, \dots, x - 1$ , но еще не выложил карту с числом  $x$ .

Игра заканчивается, когда нельзя сделать ни одного из вышеперечисленных ходов. Цель игры состоит в том, чтобы выложить на стол как можно больше карт.

Петя подсмотрел, в каком порядке лежат карты в колоде, и теперь хочет сделать ходы так, чтобы максимизировать число выложенных карт. Помогите ему выяснить, какое максимальное число карт ему удастся выложить.

### Формат входных данных

Входной файл содержит несколько тестовых примеров. В первой строке находится целое число  $T$  — количество тестовых примеров ( $1 \leq T \leq 10^5$ ). Следующие  $2T$  строк содержат описание тестовых примеров.

Описание каждого тестового примера состоит из двух строк. В первой строке находятся целые числа  $n$ ,  $m$  и  $k$  — количество карт в колоде, максимальное число, которое может быть написано на карте, и максимальное количество карт в руке ( $1 \leq n, m \leq 10^5$ ,  $1 \leq k \leq n$ ).

Во второй строке находятся  $n$  целых чисел  $a_i$  — числа, написанные на картах, в том порядке, в котором они лежат в колоде, начиная с самой верхней карты ( $1 \leq a_i \leq m$ ).

Сумма  $n$  во всех тестовых примерах не превосходит  $10^5$ .

### Формат выходных данных

Для каждого тестового примера в отдельной строке выведите единственное целое число — максимальное число карт, которое можно выложить.

### Примеры

game.in	game.out
3	2
4 3 1	0
3 2 1 2	4
1 2 1	
2	
5 5 2	
4 2 1 4 3	

В третьем примере следует играть следующим образом. Исходно у Пети в руках 4 и 2. Петя сбрасывает 4, берет из колоды 1. Теперь у него в руках 2 и 1. Петя выкладывает 1, затем 2. Теперь у него в руке нет карт. Он берет из колоды 4, затем берет из колоды 3. Теперь у него в руке 4 и 3, он выкладывает 3 и затем выкладывает 4.

## Задача Е. Гипотеза об обобщенном коне

Имя входного файла: `knight.in`  
Имя выходного файла: `knight.out`

Шахматный аналитик Миша занимается исследованием перемещений произвольных шахматных фигур по доскам произвольного размера. Вот несколько определений из его последней монографии.

Обобщенной шахматной доской  $m \times n$  будем называть прямоугольную доску, которая имеет  $m$  столбцов и  $n$  рядов. Клетка обобщенной доски задается парой целых чисел  $(c, r)$ , где  $c$  изменяется от 1 до  $m$  и задает номер столбца, а  $r$  от 1 до  $n$  и задает номер ряда.

Обобщенный конь — это вымышленная шахматная фигура, задаваемая конечным множеством возможных ходов, где каждый ход — это пара целых чисел. Если пара  $(x, y)$  входит в множество возможных ходов, то своим ходом обобщенный конь может переместиться из клетки с координатами  $(c, r)$  в клетку с координатами  $(c + x, r + y)$ , если эта клетка принадлежит доске. Например, обычный шахматный конь — это не что иное, как обобщенный конь, задаваемый множеством  $\{(2, 1), (1, 2), (-1, 2), (-2, 1), (-2, -1), (-1, -2), (1, -2), (2, -1)\}$ .

Будем говорить, что обобщенный конь *прекрасно перемещается* по прямоугольной доске  $m \times n$ , если из любой клетки этой доски до любой другой он может добраться, не выходя за границы доски. Например, обычный шахматный конь прекрасно перемещается по доске  $8 \times 8$ .

Недавно Миша выбрал числа  $a, b, c$  и  $d$  и сформулировал гипотезу: «если обобщенный конь прекрасно перемещается по доске  $a \times b$ , то он прекрасно перемещается по доске  $c \times d$ ». Помогите определить, верна ли эта гипотеза для заданных  $a, b, c$  и  $d$ , и если нет, то приведите пример обобщенного коня, который является для нее контрпримером.

### Формат входных данных

Входной файл содержит одну строку, в которой находятся четыре целых числа  $a, b, c$  и  $d$  ( $1 \leq a, b, c, d \leq 50$ ).

### Формат выходных данных

В первой строке выведите «YES», если гипотеза верна, и «NO» иначе.

Если гипотеза не верна, выведите описание обобщенного коня, который является к ней контрпримером. Во второй строке выведите количество ходов, которые может делать этот обобщенный конь, после чего выведите эти ходы, по одному на строке. Если возможных контрпримеров несколько, выведите любой.

### Примеры

<code>knight.in</code>	<code>knight.out</code>
8 8 8 2	NO 8 2 1 1 2 -1 2 -2 1 -2 -1 -1 -2 1 -2 2 -1
4 4 8 8	YES

### Пояснения к примерам

Утверждение «любой обобщенный конь, прекрасно перемещающийся по доске  $8 \times 8$ , также прекрасно перемещается по доске  $8 \times 2$ » является ложным. Обычный шахматный конь является контрпримером к нему: он прекрасно перемещается по доске  $8 \times 8$ , но не может добраться из клетки  $(1, 1)$  до клетки  $(1, 2)$  на доске  $8 \times 2$ .

А утверждение «любой обобщенный конь, прекрасно перемещающийся по доске  $4 \times 4$ , также прекрасно перемещается по доске  $8 \times 8$ » является истинным.

## Задача F. ЕГЭ

Имя входного файла: `notation.in`  
Имя выходного файла: `notation.out`

Соня — одиннадцатиклассница, и в этом году ей надо сдавать единый государственный экзамен по информатике. Она решила начать готовиться заранее и стала решать задачи из вариантов прошлых лет.

В многих заданиях требуется перевести число из одной системы счисления в другую. Соня с легкостью справляется с такими заданиями, но недавно в одном из вариантов ей попала задача, которая показалась довольно интересной: число  $x$ , заданное в десятичной системе счисления, требуется перевести в  $(-2)$ -ичную систему счисления.

Формально, запись числа в  $(-2)$ -ичной системе счисления называется набор чисел  $a_0, a_1, \dots, a_{n-1}$ , каждое из которых равно 0 или 1, причем  $n = 1$  или  $a_{n-1} \neq 0$  и выполнено равенство:

$$x = \sum_{i=0}^{n-1} a_i (-2)^i.$$

Например, 3 в  $(-2)$ -ичной системе счисления представляется набором  $(1, 1, 1)$ : действительно,  $1 \cdot (-2)^0 + 1 \cdot (-2)^1 + 1 \cdot (-2)^2 = 1 - 2 + 4 = 3$ .

В задаче предлагается перевести в  $(-2)$ -ичную систему счисления только одно число, но Соне стало интересно решение этой задачи в общем случае.

После долгих раздумий она обратилась к вам за помощью. Помогите ей перевести заданное число в  $(-2)$ -ичную систему счисления.

### Формат входных данных

В единственной строке входного файла записано одно целое число  $x$  — число, которое Соня хочет представить в  $(-2)$ -ичной системе счисления ( $-10^{18} \leq x \leq 10^{18}$ ).

### Формат выходных данных

В первой строке выходного файла выведите число  $n$  ( $n \geq 1$ ).

Во второй строке выходного файла через пробел выведите  $n$  чисел  $a_0, a_1, \dots, a_{n-1}$  — цифры числа  $x$  в  $(-2)$ -ичной системе счисления ( $0 \leq a_i \leq 1$ ).

Если существует несколько ответов, выведите любой из них.

### Примеры

<code>notation.in</code>	<code>notation.out</code>
3	3 1 1 1
-2	2 0 1

## Задача G. Генератор паролей

Имя входного файла: `pwgen.in`  
Имя выходного файла: `pwgen.out`

После ряда утечек конфиденциальной информации общественность всерьез задумалась о том, какие пароли следует использовать. Например, пароль «11111» — простой и его не следует использовать. Но как определить, хороший ли пароль?

Фондом Стандартизации Бизнес-процессов было проведено исследование, показавшее, что хороший пароль должен быть достаточно длинным и содержать символы разных видов. Исследование показало, что в хорошем пароле должно быть ровно  $n$  символов. Пароль должен состоять из заглавных и строчных букв латинского алфавита и цифр и удовлетворять следующим требованиям:

- в пароле должно быть хотя бы  $a$  заглавных букв;
- в пароле должно быть хотя бы  $b$  строчных букв;
- в пароле должно быть хотя бы  $c$  цифр;
- в пароле не должно быть двух одинаковых идущих подряд символов.

Понимая, что пользователям будет некомфортно придумывать пароль, удовлетворяющий всем этим требованиям, работники Фонда решили написать программу, призванную помочь пользователям в этом деле.

Помогите Фонду, напишите программу, которая по заданным параметрам генерирует подходящий пароль.

### Формат входных данных

В первой строке входного файла задано целое число  $n$  — требуемая длина пароля ( $1 \leq n \leq 100$ ). Во второй строке заданы неотрицательные целые числа  $a$ ,  $b$  и  $c$  — минимальное необходимое число заглавных букв, строчных букв и цифр, соответственно ( $a + b + c \leq n$ ).

### Формат выходных данных

В единственной строке выведите любой *хороший* пароль. Допустимо использовать только заглавные и строчные латинские буквы, а также цифры.

### Примеры

<code>pwgen.in</code>	<code>pwgen.out</code>
8 2 5 1	PasSw0rd

## Задача Н. Уборка снега

Имя входного файла: `snow.in`  
Имя выходного файла: `snow.out`

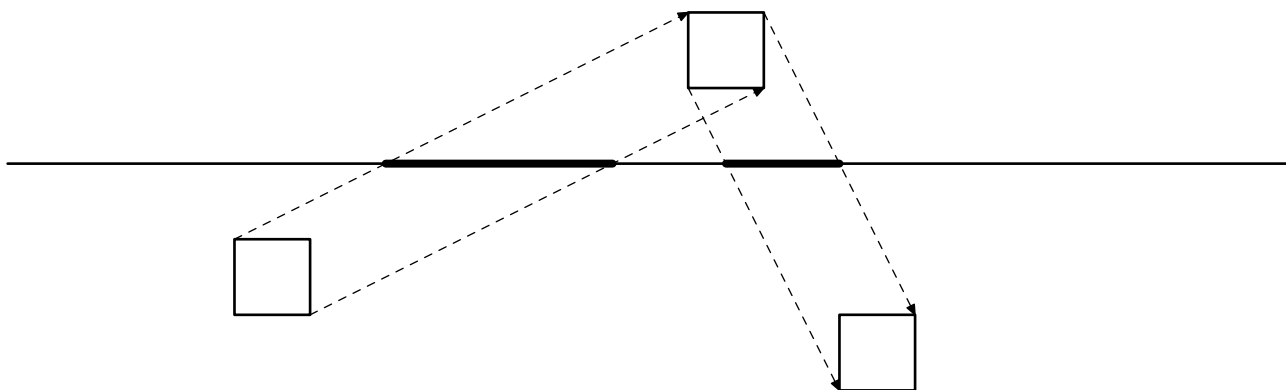
В Лайнландии есть одна очень важная дорога, имеющая форму прямой на плоскости. Зимой дорогу часто засыпает снегом, и коммунальные службы стараются оперативно ее очищать.

Лайнландские метеорологи предсказали, что сегодня ночью пройдет снегопад. Они даже в точности смогли описать его ход. Ровно в полночь над страной сформируется облако, из которого будет непрерывно падать снег. Можно считать, что облако представляет собой выпуклый многоугольник, если смотреть на него сверху.

Также метеорологи прогнозируют очень сильный и изменчивый ветер. Благодаря этому ветру траекторией движения облака будет ломаная. Движение облака за ночь можно разбить на несколько участков. Каждый участок задается вектором перемещения облака, во время перемещения вдоль участка облако движется прямолинейно, при этом каждая точка облака перемещается на заданный вектор, облако не меняет форму и не поворачивается.

Каждая точка дороги, над которой пролетит облако, будет покрыта снегом. Коммунальные службы хотят как можно быстрее расчистить дорогу от снега. Для этого они хотят знать общую длину дороги, которая окажется покрыта снегом с утра. Помогите им найти эту величину.

Рисунок показывает перемещения облака и покрытую снегом часть дороги в первом примере.



### Формат входных данных

В первой строке содержится описание дороги. Она задается двумя различными точками, которые на ней лежат. Каждая точка описывается двумя целыми числами  $x, y$  ( $-10^8 \leq x, y \leq 10^8$ ).

Далее следует описание облака в виде выпуклого многоугольника. В первой строке записано целое число  $n$  — количество вершин многоугольника ( $3 \leq n \leq 10^5$ ). В следующих  $n$  строках содержится описание положения облака в полночь, каждая строка содержит два целых числа  $x_i$  и  $y_i$  — начальное положение  $i$ -й вершины ( $-10^8 \leq x_i, y_i \leq 10^8$ ). Вершины заданы в порядке обхода многоугольника против часовой стрелки.

В следующей строке задано целое число  $m$  — количество изменений направления ветра ( $0 \leq m \leq 10^5$ ). В следующих  $m$  строках записано, как изменилось положение облака в течение очередного участка времени. Каждый участок характеризуется двумя целыми числами  $dx_i$  и  $dy_i$  — на сколько переместилось облако по каждой координате за время  $i$ -го участка ( $-10^3 \leq dx_i, dy_i \leq 10^3$ ). Гарантируется, что хотя бы одно из этих двух чисел не равно нулю.

### Формат выходных данных

Выведите одно вещественное число — общую длину дороги, которая покрыта снегом. Ответ будет считаться верным, если он имеет относительную или абсолютную погрешность не более  $10^{-9}$ .



## Пример

snow.in	snow.out
0 3 1 3 4 -4 2 -5 2 -5 1 -4 1 2 6 3 2 -4	4.5
0 0 1 1 5 1 -2 3 -2 3 -1 2 0 1 -1 4 -1 1 1 1 -7 0 0 -20	5.65685424949238019

## Задача I. Черепахи в пруду

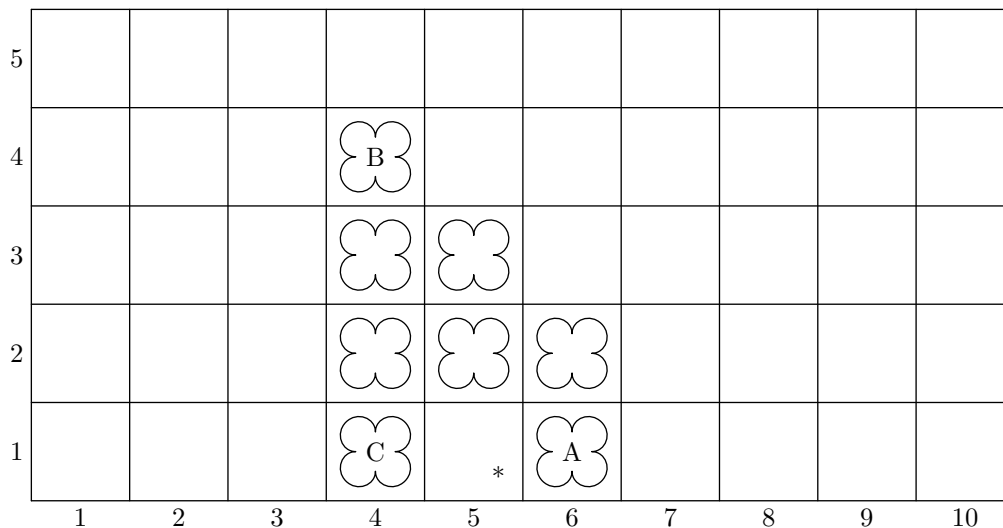
Имя входного файла: `turtles.in`  
Имя выходного файла: `turtles.out`

Пруд черепахи Тортиллы сильно изменился с момента последнего посещения Буратино. Теперь он представляет собой прямоугольное поле из  $w$  столбцов и  $h$  строк, в некоторых клетках которого расположены кувшинки. На этих кувшинках в солнечные дни греются многочисленные дети, внуки и правнуки Тортиллы.

На пруду находится набор из  $n$  *связных* кувшинок. Набор кувшинок называется связным, если от любой кувшинки можно дойти до любой другой, переходя по кувшинкам, смежным по стороне.

Черепахи очень любят ходить друг к другу в гости. Когда одна черепаха решает нанести визит другой, она строит некоторый маршрут, проходящий через смежные по стороне кувшинки и соединяющий ее кувшинку с кувшинкой своей знакомой. Однако неподвижный образ жизни черепах накладывает серьезные ограничения на возможные маршруты — перед началом движения черепаха выбирает некоторые два из четырех возможных направлений перемещения (вверх, вниз, влево, вправо) и затем может переходить с кувшинки на кувшинку только в одном из этих двух направлений.

Например, на приведенном ниже рисунке черепахе с кувшинки А, желающей нанести визит черепахе с кувшинки В, нужно выбрать направления вверх и влево, чтобы иметь возможность проложить маршрут. С другой стороны, черепаха с кувшинки С не имеет возможности добраться до кувшинки А, так как на любом пути, соединяющем их кувшинки, потребуется перемещаться по меньшей мере в трех направлениях.



Черепахи считают, что кувшинки в пруду расположены *удобно для черепах*, если черепаха может с любой кувшинки добраться до любой другой, используя для перемещения только два направления. Например, кувшинки в пруду на рисунке выше расположены неудобно, так как с кувшинки, отмеченной буквой С, нельзя добраться до кувшинки, отмеченной буквой А. Если же добавить к имеющимся кувшинкам кувшинку на клетку, отмеченную звездочкой, то кувшинки в пруду будут расположены удобно для черепах.

По итогам семейного совещания Тортилла приняла решение последовательно добавлять к  $n$  имеющимся кувшинкам  $q$  новых.

Ваша задача — до начала добавлений и после каждого добавления определить, является ли набор кувшинок удобным для черепах. Тортилла гарантирует вам, что после каждого очередного добавления набор кувшинок остается связным.

### Формат входных данных

В первой строке находятся два целых числа  $h$ ,  $w$  — количество строк и столбцов в клетчатом поле, которым является пруд ( $1 \leq h, w \leq 100\,000$ ).

В следующей строке находится целое число  $n$  — количество кувшинок в пруду в начальный момент ( $1 \leq n \leq 100\,000$ ).

В последующих  $n$  строках находится по два целых числа  $r_i, c_i$  — для каждой кувшинки заданы номера строки и столбца, на пересечении которых она находится ( $1 \leq r_i \leq h, 1 \leq c_i \leq w$ ).

В следующей строке находится целое число  $q$  — количество кувшинок, которые собирается добавить Тортилла ( $0 \leq q \leq 100\,000$ ).

В последующих  $q$  строках в аналогичном формате заданы пары целых чисел  $nr_i, nc_i$ , обозначающие соответственно номер строки и номер столбца клетки с очередной добавленной кувшинкой ( $1 \leq nr_i \leq h, 1 \leq nc_i \leq w$ ).

Гарантируется, что никакие две кувшинки во входных данных не совпадают. Гарантируется, что исходно и после каждого добавления набор кувшинок является связным.

## Формат выходных данных

Выведите  $q+1$  строку, каждая из которых представляет собой либо слово «YES», либо слово «NO», в зависимости от того, является ли на очередной момент времени система кувшинок удобной для черепах или нет. Первая строка должна содержать ответ для начального расположения кувшинок, каждая из последующих должна содержать ответ после очередного добавления кувшинки.

## Примеры

turtles.in	turtles.out
5 10	NO
8	YES
1 4	YES
2 4	NO
2 5	YES
2 6	
1 6	
3 5	
3 4	
4 4	
4	
1 5	
2 7	
3 7	
3 6	
3 3	YES
5	NO
1 1	NO
1 2	NO
1 3	YES
2 3	
3 3	
4	
2 1	
3 2	
3 1	
2 2	

## Задача J. Починка забора

Имя входного файла: `wall.in`  
Имя выходного файла: `wall.out`

Дед Афанасий решил подлатать забор вокруг своего деревенского участка. В прошлом году он как раз строил сарай, так что доски для ремонта у него остались.

Забор состоит из  $n$  сегментов, каждый из которых представляет собой доску высотой  $a_i$ . У деда есть тележка, на которой лежит стопка из  $m$  досок, про каждую из которых известна ее длина  $b_i$ .

Дед Афанасий идет вдоль забора и катит перед собой тележку с досками. Если он хочет увеличить высоту текущего сегмента, он может взять доску с тележки и прибить ее сверху. Тогда новая высота сегмента будет равна сумме изначальной высоты сегмента и длины прибитой доски. Дед не хочет прибивать больше одной доски к каждому сегменту забора, чтобы сохранить его прочность.

Собираясь увеличить высоту сегмента забора, Афанасий поступает следующим образом. Он либо использует для увеличения сегмента верхнюю доску с тележки, либо выкидывает одну или несколько верхних досок с тележки и использует следующую доску. Силы у Афанасия уже не те, поэтому он никогда не возвращается назад вдоль забора и никогда не подбирает выкинутые ранее доски.

Перед началом работы дед задумался, какую максимальную высоту может иметь забор после починки. Высотой забора Афанасий считает высоту самого низкого сегмента забора.

Помогите деду Афанасию узнать, какую максимальную высоту забора он сможет получить.

### Формат входных данных

В первой строке входного файла находится целое число  $n$  — количество сегментов в заборе ( $1 \leq n \leq 10^5$ ). Во второй строке содержатся  $n$  целых чисел  $a_1, a_2, \dots, a_n$  — высоты сегментов забора, перечисленные в том порядке, в котором мимо них пройдет дед Афанасий ( $1 \leq a_i \leq 10^8$ ).

В третьей строке находится целое число  $m$  — количество досок на тележке ( $1 \leq m \leq 10^5$ ). В четвертой строке содержатся  $m$  целых чисел  $b_1, b_2, \dots, b_m$  — длины досок на тележке, начиная с верхней ( $1 \leq b_i \leq 10^8$ ).

### Формат выходных данных

В первую строку выходного файла выведите два целых числа  $h$  и  $k$  — максимальную возможную высоту забора и количество досок, которые деду следует использовать при починке. В следующих  $k$  строках выведите по два целых числа  $x_i$  и  $y_i$ , которые означают, что к  $x_i$ -му сегменту забора деду следует прибить доску с номером  $y_i$ .

Если существует несколько способов починить забор требуемым образом, выведите любой из них.

### Примеры

<code>wall.in</code>	<code>wall.out</code>
3 10 5 10 1 5	10 1 2 1
6 2 5 4 1 7 5 7 2 3 1 3 2 4 6	5 3 1 2 3 4 4 7

## Задача К. Прогулка по зоопарку

Имя входного файла: `zoo.in`  
Имя выходного файла: `zoo.out`

Андрей Сергеевич работает сторожем в зоопарке. Однажды утром он пришел на работу и обнаружил анонимную записку, в которой сообщалось, что ночью хулиганы поменяли таблички на двух клетках. Андрей Сергеевич должен срочно решить проблему и поменять таблички обратно. К сожалению, Андрей Сергеевич хорошо разбирается в охране зоопарка, но не столь силен в зоологии, так что понять по внешнему виду животного, что это за зверь, он не в силах.

К счастью, на каждой табличке помимо наименования животного написан также порядковый номер. Номера на всех табличках разные. Обойдя зоопарк, Андрей Сергеевич составил его карту. В зоопарке есть  $n$  перекрестков, соединенных  $m$  дорожками. Чтобы встречные потоки посетителей не мешали друг другу наслаждаться посещением зоопарка, по каждой дорожке разрешено перемещаться ровно в одном направлении, от начала дорожки к ее концу. При этом от одного перекрестка до другого идет максимум одна дорожка, не существует дорожки, которая начинается и заканчивается на одном и том же перекрестке. Рядом с каждой дорожкой расположена ровно одна клетка с животным.

Перекрестки в зоопарке пронумерованы от 1 до  $n$ , Андрей Сергеевич для каждой дорожки записал номер перекрестка, на котором она начинается, номер перекрестка, на котором она заканчивается, и номер на табличке, в настоящий момент находящейся на клетке с животным около этой дорожки.

Около входа в зоопарк расположен список тематических маршрутов, рекомендуемых сотрудниками зоопарка. Каждый тематический маршрут задается начальным перекрестком, конечным перекрестком и последовательностью дорожек, по которым следует пройти, чтобы изучить соответствующих животных. Первая дорожка в маршруте начинается на его начальном перекрестке, каждая следующая дорожка начинается на перекрестке, где заканчивается предыдущая, а последняя дорожка завершается на конечном перекрестке маршрута. Каждая дорожка задается номером таблички на клетке с животным, расположенной около соответствующей дорожки. Никакой маршрут не проходит через один перекресток два раза. По каждой дорожке в зоопарке проходит хотя бы один тематический маршрут.

Андрею Сергеевичу тут же пришла в голову гениальная мысль: используя информацию о тематических маршрутах, он может выяснить, какие две таблички хулиганы поменяли местами. Но, к сожалению, задача оказалась для него слишком сложной. Помогите сторожу выяснить, какие две таблички поменяли местами, чтобы он успел восстановить порядок до открытия зоопарка.

Выясните, какие две таблички можно поменять местами, чтобы все тематические маршруты в зоопарке действительно существовали. Гарантируется, что это можно сделать.

### Формат входных данных

В первой строке содержится два целых числа  $n$  и  $m$  — количество перекрестков и количество дорожек, соответственно ( $1 \leq n, m \leq 100\,000$ ). Все перекрестки пронумерованы от 1 до  $n$ .

Следующие  $m$  строк содержат по три целых числа  $a_i$ ,  $b_i$  и  $c_i$  — номер перекрестка, где начинается очередная дорожка, номер перекрестка, где она заканчивается, и номер на табличке на расположенной на этой дорожке клетке ( $1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq c_i \leq m$ ). Все номера на табличках различны.

В следующей строке содержится целое число  $k$  — количество тематических маршрутов по зоопарку ( $1 \leq k \leq 100\,000$ ).

Следующие  $2k$  строк описывают маршруты. Первая строка описания маршрута содержит три целых числа  $l_i$ ,  $s_i$  и  $t_i$  — число дорожек в маршруте, номер перекрестка, где маршрут начинается и номер перекрестка, где он заканчивается ( $1 \leq l_i \leq n, 1 \leq s_i, t_i \leq n, s_i \neq t_i$ ). Во второй строке содержится  $l_i$  чисел — последовательность номеров на табличках на клетках, вдоль которых проходит этот маршрут.

Суммарная длина всех маршрутов не превышает 100 000. Никакой маршрут не проходит через один перекресток два раза. По каждой дорожке в зоопарке проходит хотя бы один тематический маршрут.

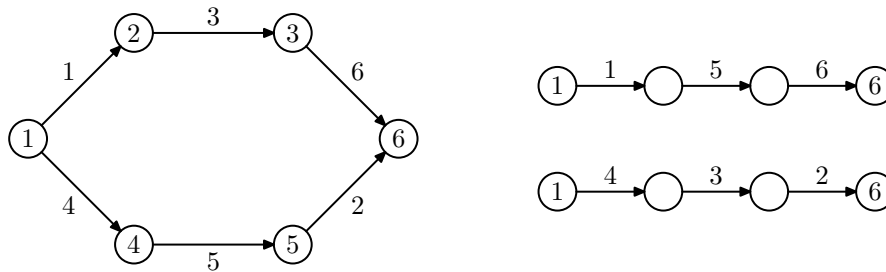
## Формат выходных данных

Выведите два различных числа — номера табличек на клетках, которые Андрей Сергеевич должен поменять местами, чтобы все тематические маршруты снова существовали. Если существует несколько решений, то выведите любое из них.

### Пример

zoo.in	zoo.out
6 6	3 5
1 2 1	
2 3 3	
3 6 6	
1 4 4	
4 5 5	
5 6 2	
2	
3 1 6	
1 5 6	
3 1 6	
4 3 2	

Карта зоопарка из примера и тематические маршруты приведены на следующем рисунке.



Видно, что если поменять местами таблички на клетках с номерами 3 и 5, то карта приобретет следующий вид и оба маршрута действительно будут существовать на карте.

