



## Задача B. Colored Ring (Division 1 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 3 seconds  
Ограничение по памяти: 256 MiB

На столе лежит кольцо,  $k$  секторов которого раскрашены в попарно различные цвета. Сектора пронумерованы последовательно по направлению часовой стрелки, и  $i$ -й сектор имеет  $i$ -й цвет. Кенга разрешила крошке Ру написать на кольце  $n$  различных целых чисел от 1 до  $n$ . Причем, число  $i$  должно быть записано или на секторе цвета  $x_i$ , или на любом другом, который не дальше  $a_i$  секторов по направлению против часовой стрелки или не дальше  $b_i$  секторов по часовой.

Кенга говорит, что числа записаны правильно, если все числа записаны только там, где она разрешила, на каждом секторе написано не более одного числа, и числа  $1, 2, \dots, n$ , которые написал крошка Ру, идут по часовой стрелке.

Теперь крошка Ру хочет знать, сколько у него различных способов испортить разноцветное кольцо числами, записывая их так, как сказала Кенга. Обратите внимание: для того, чтобы испортить один сектор, неважно, какое число на нем писать.

### Формат входного файла

В первой строке записано два целых числа  $n$  and  $k$  ( $1 \leq n \leq 15, 1 \leq k \leq 60, n \leq k$ ). В следующих  $n$  строках записано по три целых числа  $x_i, a_i, b_i$  ( $1 \leq x_i \leq k; 0 \leq a_i, b_i; a_i + b_i < k$ ). Последовательность  $x_i$  строго возрастает.

### Формат выходного файла

Выведите количество способов испортить разноцветное кольцо.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
1 5 1 2 1	4
3 8 4 0 3 5 0 3 6 0 0	3
3 7 2 2 4 4 1 3 6 1 5	35

## Задача C. Christopher Robin is Learning Sorting Permutations (Division 1 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 4 seconds (*7 seconds for Java*)  
Ограничение по памяти: 256 MiB

```
// WRITTEN BY ME AND MR CORMEN HELPD
int psort(int L, int R) {
    if (L >= R)
        return 0;

    int i = L;
    int j = R;
    int x = Random.next();
    int x0 = A[x];

    assert(L <= x && x <= R);

    while (i < j) {
        while (A[i] < A[x]) i++;
        while (A[j] > A[x]) j--;
        if (i <= j) {
            if (i == x)
                x = j;
            else if (j == x)
                x = i;
            swap(A[i], A[j]);
            i++;
            j--;
        }
    }

    swap(A[x], A[x0]);

    return R - L + psort(L, x0 - 1) + psort(x0 + 1, R);
}
```

(В последней строке рекурсивный вызов функций происходит в порядке слева направо.)

Вам дано число  $n$  и последовательность чисел, которую возвращает `Random.next()`. Найдите такой массив  $A$ , что `psort(1, n)` будет возвращать наибольшее возможное значение. Массив  $A$  должен быть перестановкой  $n$  целых чисел от 1 до  $n$ . Обратите внимание, что сортировка может быть неуспешной из-за строки:

```
assert(L <= x && x <= R)
```

### Формат входного файла

В первой строке записано одно целое число  $n$  ( $1 \leq n \leq 50$ ). Во второй строке записано  $n$  целых чисел от 1 до  $n$ , разделенных пробелами. Это числа, которые возвращают последовательные вызовы функции `Random.next()`.

## Формат выходного файла

Если такого массива **A** не существует, выведите «No solution». Иначе, выведите в первой строке «Solution exists» и сам массив **A** на второй строке. Если существует несколько таких массивов, выведите любой из них.

## Примеры

input.txt	output.txt
3 1 2 3	Solution exists 1 2 3
7 1 7 1 7 1 7 1	No solution

## Задача D. Clever Plan

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds (*7 seconds for Java*)  
Ограничение по памяти: 256 MiB

Кролик подготовил Очень Умный План для Пуха, чтобы обмануть Слонопотама. Представьте себе  $n$  горшков с медом, которые Пух распределяет между собой и Слонопотамом таким образом, что каждый из них получает по крайней мере по одному горшку, и все такие распределения равновероятны. (Горшки неразличимы, так что таких распределений всего  $n - 1$ .)

Теперь Пух и Слонопотам начинают играть. Игра состоит из раундов, и она заканчивается, если у Слонопотама или у Пуха больше нет горшков с медом. Каждый раунд играется по невероятно сложному плану Кролика, поэтому все, что можно сказать о раунде, это всего лишь три числа:  $p$ ,  $q$ ,  $r$  ( $p + q + r = 1$ ). Во время раунда происходит одно из следующих событий: или Слонопотам получает один горшок от Пуха с вероятностью  $p$ , или Пух получает от Слонопотама один горшок с вероятностью  $q$ , или все остается как прежде с вероятностью  $r$ . И так каждый раунд.

Допустим, было сыграно  $i$  раундов, игра еще не закончилась, а распределение горшков между Пухом и Слонопотамом оказалось точно таким же, как и до начала первого раунда. Какова вероятность такого события?

### Формат входного файла

В первой строке содержится одно целое число  $T$  — количество тестов ( $1 \leq T \leq 200$ ).

Каждый тест состоит из трех строк. Первая строка каждого теста содержит одно целое число  $n$  — количество горшков с медом ( $3 \leq n \leq 26$ ). Вторая строка  $t$ -го теста содержит три числа:  $p_t$ ,  $q_t$ ,  $r_t$  ( $0.3 \leq p_t, q_t, r_t < 1$ ,  $p_t + q_t + r_t = 1$ ). Вероятности даны с точностью не более двух знаков после запятой. В третьей строке записано целое число  $k_t$  ( $1 \leq k_t \leq 1600$ ) — максимальное число раундов.

### Формат выходного файла

Для каждого теста выведите  $k_t$  строк;  $i$ -я из них должна содержать число — вероятность события, описанного в условии (после  $i$ -го раунда). Числа должны быть выведены с относительной погрешностью не более  $10^{-3}$ .

### Пример

input.txt	output.txt
1	4.000000e-01
3	2.500000e-01
0.3 0.3 0.4	1.720000e-01
5	1.201000e-01
	8.404000e-02

## Задача E. Visiting (Division 1 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Винни-Пух очень любит ходить в гости. В Лесу есть несколько тропинок, по которым он может ходить из одного дома в другой. Пух хочет, чтобы по этим тропинкам можно было добраться от каждого дома до любого другого.

По каждой из тропинок Винни-Пух ходит только в одну сторону. Это потому что он всегда бормочет себе под нос песенки, когда идет, и не может бормотать их задом наперед. Так что Пуху потребуется немного новых тропинок.

### Формат входного файла

В первой строке записано два целых числа  $n$  и  $m$  ( $1 \leq n \leq 500$ ,  $0 \leq m \leq 800$ ) — количество домов и количество существующих тропинок соответственно.

В следующих  $m$  строках содержится описание тропинок. Каждое описание состоит из двух чисел  $a_i$  и  $b_i$  ( $a_i \neq b_i$ ), что означает, что Пух ходит по этой тропинке от дома  $a_i$  к дому  $b_i$ . Дома пронумерованы от 1 до  $n$ . Все пары  $(a_i, b_i)$  различны.

### Формат выходного файла

На первой строке выведите число  $k$  — наименьшее возможное число новых тропинок.

В следующих  $k$  строках выведите описания новых тропинок. Каждое описание должно следовать тому же формату, что и во входном файле. Ответ должен быть лексикографически минимальным.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
3 2 1 2 1 3	2 2 1 3 1
4 2 1 2 3 4	2 2 3 4 1

### Note

Последовательность пар  $u_1, u_2, \dots, u_k$  лексикографически меньше, чем последовательность пар  $v_1, v_2, \dots, v_k$ , если существует такой индекс  $i$ ,  $1 \leq i \leq k$ , что  $u_i < v_i$  и  $u_j = v_j$  для всех  $j < i$ . Здесь все пары также сравниваются лексикографически.

Пара  $(a_1, b_1)$  лексикографически меньше, чем пара  $(a_2, b_2)$ , если  $a_1 < a_2$  или выполняются два условия:  $a_1 = a_2$  и  $b_1 < b_2$ .

## Задача F. Repeating $b$ -ary

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Пытаясь выбраться из дома Кролика через нору, Пух в ней застрял. Кристофер Робин сказал, что придется подождать, пока Пух не похудеет. Чтобы поддержать и утешить медвежонка, Кристофер Робин предложил ему записать  $b$ -ичное представление числа  $\frac{1}{x}$ . К тому времени, когда Винни-Пух закончит его записывать, он, естественно, похудеет. А сколько времени это займет, зависит от количества чисел в непериодической части и количества чисел в периодической части  $b$ -ичного представления числа. Вам дано  $x$ , найдите эти числа.

### Формат входного файла

В единственной строке записаны числа  $x$  и  $b$  ( $2 \leq x \leq 10^{12}$ ,  $2 \leq b \leq 10^{18}$ ), где  $b$  — основание системы счисления, в которой Пуху нужно записать число.

### Формат выходного файла

Выведите минимальные возможные длины непериодической и периодической частей.

### Примеры

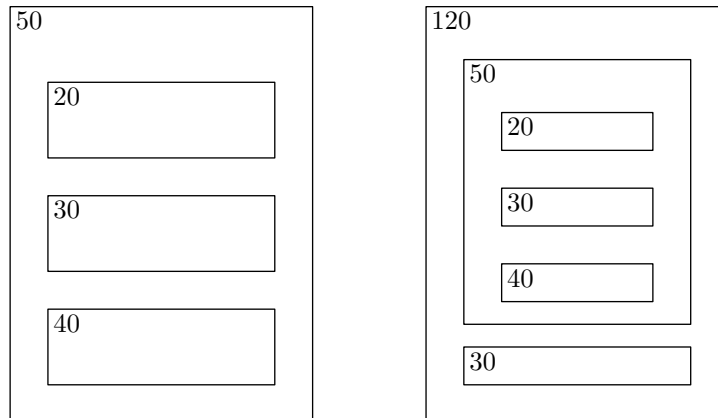
<code>input.txt</code>	<code>output.txt</code>
2 10	1 0
3 10	0 1
2 2	1 0
3 2	0 2

## Задача G. Christopher Robin is Learning Object-oriented Programming (Division 1 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 3 seconds (6 seconds for Java)  
Ограничение по памяти: 256 MiB

В этой задаче вам потребуется сравнить все объекты друг с другом.

Каждый объект имеет скалярную величину и список ссылок на другие объекты (возможно пустой). Порядок ссылок в списке **важен**. Объекты можно изобразить следующими диаграммами:



Здесь первый объект имеет скалярную величину 50 и ссылается на три объекта. Второй объект ссылается на первый объект и на объект со скалярной величиной 30.

Можно сказать, что объекты равны, если их диаграммы неотличимы. Конечно, такие диаграммы нельзя изобразить на бумаге при наличии в объектах циклических ссылок, но тем не менее диаграммы таких объектов можно сравнивать.

### Формат входного файла

В первой строке входного файла записано число  $n$  ( $2 \leq n \leq 10^5$ ) — количество объектов. В следующих  $n$  строках описаны сами объекты.

Каждое описание начинается с целого числа  $s$  ( $0 \leq s \leq 10^5$ ) — скалярной величины объекта. Далее следует число  $m$  ( $0 \leq m \leq 10$ ) и  $m$  чисел — список ссылок. Каждая ссылка является номером объекта. Объекты пронумерованы от 1 до  $n$  в том же порядке, в каком они следуют во входном файле.

### Формат выходного файла

Выведите ровно  $n$  строк.

В  $i$ -ой строке должен быть записан номер класса, которому принадлежит  $i$ -ый объект. Равные объекты должны принадлежать одному классу, различные — разным. Номера классов должны быть целыми числами от 1 до  $2^{63} - 1$ , включительно.



## Пример

input.txt	output.txt
8	10
50 3 3 4 5	12
120 2 1 4	3
20 0	4
30 0	5
40 0	10
50 3 3 4 5	12
120 2 6 4	11
120 2 4 6	

## Задача H. Expedition (Division 1 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Винни-Пух брел по Лесу, собираясь повидать своего друга Кристофера Робина и выяснить, не позабыл ли он о том, что на свете существуют медведи. Это был день, когда Кристофер Робин готовился к экспедиции на Северный Полюс. Кристофер Робин попросил Пуха сказать всем остальным, чтобы они тоже собирались и не забыли провизию. Пух уже успел поговорить с Кроликом и теперь торопился к дому Пятачка.

— Эй, Пятачок! — взволнованно сказал Пух. — Мы все отправляемся в экспедицию. Все, все! И берем про... Покушать. Мы должны что-то открыть.

Поначалу, Пятачка напугало это *что-то*, но потом он сказал: «Если с нами пойдет Кристофер Робин, я тогда вообще ничего не боюсь!»

Спустя немного времени все были в сборе, и экспедиция началась. В самом конце, растянувшись длинной цепочкой, шли все Родные и Знакомые Кролика.

— Я их не приглашал, — небрежно объяснил Кролик, — они просто взяли и пришли. Они всегда так. Они могут идти в конце, позади Иа.

Иа начал жаловаться на Родных и Знакомых, потому что они действуют на нервы. Иа уже был готов отказаться от всей экспедиции.

Перед Кристофером Робинем встала задача. Всего собралось в экспедицию  $n$ . Кристофер Робин хочет, чтобы  $k$  из них пошли в экспедицию не смотря ни на что (он уже решил, кто именно). Также ему хочется, чтобы экспедиционная группа была полной: чтобы не было никакой возможности добавить еще одного, двух или больше так, чтобы никто не начал жаловаться. Так кто же пойдет в конце-концов открывать Северный Полюс?

### Формат входного файла

В первой строке записаны два целых числа  $n$  and  $m$  ( $1 \leq n \leq 150\,000$ ,  $0 \leq m \leq 150\,000$ ). В следующих строках записаны требования двух различных типов. Каждое требование состоит из трех целых чисел  $t_i$ ,  $a_i$  and  $b_i$  ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ). Если  $t_i = 1$ , то  $a_i$  ни за что не пойдет в экспедицию, если надоедливый  $b_i$  собрался пойти. (Кристофер Робин не смог вспомнить всех имен Родных и Знакомых Кролика, так что он просто пронумеровал всех присутствующих.) И если  $t_i = 2$ , то  $a_i$  никуда не пойдет без своего друга  $b_i$ .

Все жители Леса довольно дружные, поэтому количества требований с  $t_i = 1$  не превосходит 40. Все пары  $(a_i, b_i)$  различны.

Следующая строка содержит целое число  $k$  ( $0 \leq k \leq n$ ). В случае, если  $k$  положительно, то последняя строка содержит  $k$  целых чисел — те, кого Кристофер Робин хочет взять обязательно.

### Формат выходного файла

В первой строке выведите единственное целое число  $s$  — количество членов экспедиции. Во второй строке выведите  $s$  чисел, разделенных пробелами, попарно различных, — кто пошел в экспедицию. Принимается любой ответ, в котором группа является полной. Гарантируется, что хотя бы один такой ответ существует.

## Примеры

input.txt	output.txt
3 2 1 1 2 1 2 3 1 3	2 1 3
3 3 2 1 2 2 1 3 1 2 3 0	1 2

## Задача I. Wrong Sort of Bees

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds (*5 seconds for Java*)  
Ограничение по памяти: 256 MiB

Пытаясь достать немного меда с самой верхушки огромного дуба, Пух держал в своих лапах воздушный шар и пел:

*Я Тучка, Тучка, Тучка,  
А вовсе не медведь,  
Ах, как приятно Тучке  
По небу лететь!  
Ах, в синем-синем небе  
Порядок и уют —  
Поэтому все Тучки  
Так весело поют!*

Ветер дул в сторону дерева, и Пух двигался вдоль прямой на постоянной высоте и с постоянной скоростью  $V_{Poh}$  метров в минуту. К сожалению, направление не было достаточно точным, и он пролетел мимо дерева.

Внезапно, Пух заметил пчелу на расстоянии  $R$  метров прямо перед собой по направлению своего движения, которая тут же начала летать вокруг Пуха на той же постоянной высоте, что и он. Пух обратил внимание (со своей точки наблюдения), что пчела ведет себя подозрительно: она летает вокруг него с постоянной угловой скоростью  $\Omega_{bee}$  оборотов в минуту. Более того, расстояние между пчелой и Пухом уменьшается с постоянной скоростью  $V_{bee}$  метров в минуту.

Когда пчела села на Пуха, он крепко задумался. Какова длина траектории пчелы? Это так важно, потому что Пух знает, что пчелы не могут летать слишком далеко. А если они все-таки летают так далеко, то это какие-то неправильные пчелы, и они, наверно, делают неправильный мед.

В этой задаче размерами Пуха и пчелы можно пренебречь.

### Формат входного файла

В первой строке записано целое число  $T$  ( $1 \leq T \leq 100$ ) — количество тестов.

Каждый тест состоит из единственной строки, содержащей четыре целых числа:  $R, V_{Poh}, V_{bee}, \Omega_{bee}$ . Гарантируется, что  $1 \leq R, V_{Poh}, V_{bee}, \Omega_{bee} \leq 100$ .

### Формат выходного файла

Выведите  $T$  строк, в каждой из которых содержится ответ на соответствующий тест — длина траектории пчелы. Числа должны быть выведены с абсолютной или относительной погрешностью не более  $10^{-4}$ .

### Пример

<code>input.txt</code>	<code>output.txt</code>
1	3.36757
1 1 1 1	

## Задача J. Maximal Sum (Division 1 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

— Знаешь, Пятачок, — сказал Пух. — Кристофер Робин складывает числа без переноса в старшие разряды.

— Неужели он пишет что-то вроде  $5 + 5 = 0$ ,  $19 + 3 = 12$  and  $75 + 25 = 90$ ?

— Да. И он называет это *поразрядной суммой*. Он просто суммирует пары соответствующих цифр по модулю 10. Интересно, какое наибольшее число он сможет получить, складывая числа только из моего списка?

### Формат входного файла

На первой строке Винни-Пух написал единственное целое число  $n$  ( $1 \leq n \leq 10^5$ ). Немного подумав, он написал еще  $n$  строчек и отнес свой список Кристоферу Робину. Таким образом, на строке с номером  $i + 1$  оказалось записано целое число  $x_i$  ( $0 < x_i < 10^6$ ).

### Формат выходного файла

В первой строке выведите одно целое число  $a$  — максимальную поразрядную сумму, которую может получить Кристофер Робин. Во второй строке выведите целое число  $m$  — сколько чисел потребовалось Кристоферу Робину. Но  $0 \leq m \leq 108$ , потому что у Винни-Пуха в голове опилки и он не сможет проверить ответ, если чисел будет больше, чем сто восемь. В третьей строке выведите  $m$  чисел — числа, поразрядная сумма которых равна  $a$ . Помните, что каждое из этих чисел должно быть одним из  $x_i$  из списка Винни-Пуха.

Если нет такого максимального ответа, который Винни-Пух сможет проверить, просто выведите  $a$  на первой строке и  $-1$  на второй.

### Примеры

input.txt	output.txt
1 1	9 9 1 1 1 1 1 1 1 1 1
2 12 13	99 9 12 12 12 12 12 12 12 12 13
2 15 5	95 9 15 15 15 15 15 15 15 15 15

## Задача К. Leapfrog (Division 1 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Все Родственники и Знакомые Кролика играют в чехарду. Сегодня,  $n$  из них выстроились в одну прямую и начали играть. А играют они следующим образом:

- 1) Один из Родственников и Знакомых (назовем его  $X$ ) выбирает другого ( $Y$ ), чтобы прыгнуть через него.
- 2)  $X$  прыгает через  $Y$ -а таким образом, что он остается на той же прямой, что и раньше, а расстояние между  $X$  и  $Y$  становится таким же, как и до прыжка.

Родственники и Знакомые Кролика достаточно малы, чтобы легко помещаться в одном и том же месте, хоть все сразу.

### Формат входного файла

В первой строке записано целое число  $n$  ( $1 \leq n \leq 100$ ).

Во второй строке записано  $n$  чисел  $a_i$ , разделенных пробелами — начальные положения Родственников и Знакомых на прямой. То есть,  $i$ -й из них стоит в позиции  $a_i$ .

В третьей строке записано  $n$  чисел, разделенных пробелами — желаемое размещение Родственников и Знакомых. Текущее размещение считается желаемым, если возможно переставить числа в текущем размещении таким образом, что список чисел-позиций в нем совпадет со списком позиций в желаемом размещении.

Все числа целые, от 1 до 100 включительно.

### Формат выходного файла

Выведите «No», если Родственники и Знакомые Кролика не смогут встать в желаемое размещение, играя в чехарду.

Иначе, выведите «Yes», а затем прыжки, которые нужно выполнить. (Можно ничего не выводить, если необходимости в прыжках нет.) Каждая следующая строка должна содержать пару чисел  $X$  и  $Y$  для каждого прыжка  $X$ -а через  $Y$ -а (имена Родственников и Знакомых: 1, 2, ...,  $n$ ).

Количество прыжков не должно превышать  $5 \times 10^5$ , и никто не должен вставать в позицию, которая находится от позиции 0 дальше, чем  $10^8$  единиц.

### Примеры

input.txt	output.txt
2 1 2 5 6	Yes 1 2 2 1 1 2 2 1
2 1 2 1 3	No

## Задача L. Shops and Balloons (Division 2 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 12 seconds  
Ограничение по памяти: 256 MiB

В графстве, в котором живёт Кристофер Робин,  $N$  городов ( $N \leq 5,000$ ), при этом каждый город может иметь торговые связи с другими городами. Всего имеется  $T$  двусторонних прямых маршрутов ( $0 \leq T \leq 25 \cdot 10^6$ ).

Для каждого прямого торгового маршрута между городами  $x$  и  $y$  определена стоимость перевозки между этими городами  $C(x, y)$ , причём  $C(x, y) \geq 0$ ,  $C(x, y) \leq 10^4$  и  $C(x, y) = C(y, x)$ .

В  $K$  из этих  $N$  городов ( $1 \leq K \leq N$ ) есть интернет-магазины, в которых можно заказать воздушные шарики соответствующей расцветки.

Цена одного шарика в городе  $x$  равна  $P_x$  ( $0 \leq P_x \leq 10^4$ ).

Требуется найти минимальную сумму затрат, необходимую для того, чтобы купить и доставить один воздушный шарик в заданный город  $D$  ( $1 \leq D \leq N$ ). Если шарик покупается в городе  $D$ , то плата за доставку равна нулю.

### Формат входного файла

В первой строке входного файла задано целое число  $N$  — количество городов. Города пронумерованы от 1 до  $N$ . Во второй строке задано число  $T$  — количество прямых маршрутов. Последующие  $T$  строк описывают сами маршруты. Каждая строка содержит 3 целых числа  $x$   $y$   $C(x, y)$ , показывающие, что данный маршрут соединяет города  $x$  и  $y$ , а затраты на перевозку шарика по этому маршруту равны  $(x, y)$ . Следующая строка содержит целое число  $K$  — количество городов, в которых можно купить шарики. В последующих  $k$  строках заданы по два целых числа  $z$  и  $P_z$ , обозначающие, что цена шарика при покупке в городе  $z$  равна  $P_z$ . Последнее число содержит целое число  $D$  — номер города, в который требуется доставить шарик.

### Формат выходного файла

Выведите одно число — минимальную сумму затрат, необходимую для покупки и доставки в пункт  $D$  одного воздушного шарика.

### Пример

input.txt	output.txt
3	6
3	
1 2 4	
2 3 2	
1 3 3	
3	
1 14	
2 8	
3 3	
1	

## Задача M. Christopher Robin is Learning Bar Graphs (Division 2 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Вам заданы  $N$  целых неотрицательных величин  $X_1, \dots, X_N$ . Требуется построить отображающую их гистограмму (клеточную диаграмму), столбцы которой составлены из символов '#', высота максимального столбца равна  $H$  символов, а остальные столбцы отмасштабированы пропорционально так, что  $i$ -й столбец имеет высоту  $k$  символов тогда и только тогда, когда отмасштабированное значение  $X_i$  не менее  $n$  и меньше  $n + 1$ .

### Формат входного файла

Входной файл состоит из одной строки, начинающейся с двух целых чисел  $H$  и  $N$  (максимальная высота столбца и количество величин). Далее заданы сами величины —  $N$  целых чисел  $X_1, \dots, X_N$ , причём как минимум одно из них не равно нулю ( $1 \leq H \leq 100, 1 \leq N \leq 100, 0 \leq X_i \leq 10^4$ ).

### Формат выходного файла

Выведите требуемую гистограмму в формате, указанном в примере к задаче.

### Пример

input.txt	output.txt
16 7 30 21 39 1 17 16 62	# # # # # # # # # # # # # # # # # # ### # ### ### ### ### ### ### ### ###



## Задача N. Encryption (Division 2 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Винни-Пух и Кролик решили использовать в своей переписке так называемый шифр Цезаря.

Пусть каждое сообщение состоит из заглавных и строчных латинских букв, а также знаков пунктуации. Сторона, отправляющая послание, увеличивает значение каждой буквы (с сохранением регистра) на некую заданную величину, называемую сдвигом. При этом сдвиг происходит циклически, то есть при значении сдвига 1 'a' переходит в 'b', 'b' в 'c', 'z' в 'a', 'K' в 'L' и так далее. Если символ не является буквой, то он должен остаться неизменным.

Ваша задача — по входному сообщению и сдвигу построить зашифрованное сообщение.

### Формат входного файла

В первой строке входного файла задано одно целое число  $n$  — сдвиг ( $1 \leq n \leq 10^9$ ). В последующей строке задано сообщение. Каждое сообщение состоит из не более, чем 1024 символов, коды которых не больше 126 и не меньше 32 и которые могут быть расположены на нескольких строках.

### Формат выходного файла

Выведите зашифрованное с данным значением сдвига сообщение, сохраняя соответствие между строками.

### Пример

<code>input.txt</code>	<code>output.txt</code>
27	Uijt jt @ Uftu Nfttbhf!
This is @ Test Message!	Op QjHmFu dbo sfbe ju!
No PiGlEt can read it!	Bsf zpv tvsf uibu
Are you sure that	Dpef xpslt?
Code works?	

## Задача O. Cool numbers (Division 2 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Пятачок считает число  $n$  *интересным*, если оно может быть представлено в виде  $n = k^p$  и  $n = l^q$ , где  $k, p, l, q$  — целые положительные числа,  $p$  и  $q$  взаимно просты и  $p, q > 1$ .

Например, 64 — интересное число, так как  $64 = 8^2$  и  $64 = 4^3$ .

Ваша задача — написать программу, которая поможет Пятачку найти все «интересные» числа в заданном диапазоне.

### Формат входного файла

В первой строке входного файла задано число  $a$  ( $1 \leq a \leq 10^8$ ), во второй — число  $b$  ( $a \leq b \leq 10^8$ ) — соответственно первое и последнее число в заданном диапазоне.

### Формат выходного файла

Выведите одно число — количество интересных чисел в заданном диапазоне.

### Пример

<code>input.txt</code>	<code>output.txt</code>
1 82	2
101 1001	1

## Задача P. Lights (Division 2 Only!)

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 MiB

Ослик Иа-Иа на свой день рождения получил забавную игрушку. Игрушка представляла собой прямоугольник из  $R$  рядов лампочек, при этом каждый из рядов содержит  $L$  лампочек. Лампочки могут находиться в одном из двух состояний: включённом или выключенном.

Занумеруем ряды снизу вверх (при этом самый нижний ряд получит номер 1, самый верхний — номер  $R$ ). Напротив каждого ряда, кроме самого верхнего, есть кнопка, которая может быть нажата.

Нажатие кнопки, расположенной напротив ряда  $k$  ( $1 \leq k < R$ ) приводит к следующим изменениям. Если в  $i$ -м столбце  $k$ -го ряда и в  $i$ -м столбце  $k + 1$ -го ряда лампочки находятся в одинаковом состоянии, то лампочка в  $i$ -м столбце  $k$ -го ряда будет выключена, если в различном — будет включена; потом рассматривается следующий столбец и так далее.

Ниже приведён пример для  $L = 4$ :

Номера столбцов	1	2	3	4
Ряд $k + 1$	on	on	off	off
Ряд $k$ до нажатия кнопки	on	off	on	off
Ряд $k$ после нажатия кнопки	off	on	on	off

По заданному изначальному состоянию игрушки определите, сколько различных состояний Иа-Иа сможет получить для нижнего ряда лампочек с помощью нажатий на кнопки.

### Формат входного файла

Первая строка входного файла содержит целое число  $R$  — количество рядов лампочек в игрушке ( $1 < R < 30$ ). Вторая строка содержит целое число  $L$  ( $1 \leq L < 8$ ) — количество лампочек в одном ряду (оно же количество столбцов). В последующих  $R$  строках задано состояние игрушки в перечислении сверху-вниз:  $i$ -е число в  $j$ -й строке равно 1, если  $i$ -я лампочка в ряду с номером  $R + 1 - j$  включена, и 0, если эта лампочка выключена.

### Формат выходного файла

Выведите одно число — количество различных состояний нижнего ряда, которые Иа-Иа сможет получить с помощью нажатий на кнопки.

### Пример

<code>input.txt</code>	<code>output.txt</code>
4 3 0 0 1 0 1 1 1 0 1 0 0 1	4