

## Problem G. Нанобудильники

Input file:        **stdin**  
Output file:       **stdout**  
Time limit:        1 секунда  
Memory limit:     256 мегабайт

*Будильник с боем — в заданное время бьёт вас до полного пробуждения.*

---

На столе старого часовщика лежат  $n$  остановившихся нанобудильников, пронумерованных натуральными числами от 1 до  $n$ . Нанобудильники измеряют время в часах, причём в одном часе миллион минут, а каждая минута длится миллион секунд. Для отладки механизмов часовщик должен синхронизировать время на всех нанобудильниках. Для этого он передвигает стрелки *вперед* на некоторое время (возможно, нулевое). Величину такого передвижения назовем временем перевода.

Ваша задача — подсчитать *наименьшее* суммарное время перевода, необходимое для того, чтобы все нанобудильники показывали одинаковое время.

### Input

В первой строке записано единственное целое число  $n$  — количество нанобудильников ( $2 \leq n \leq 10^5$ ). В каждой  $i$ -ой из  $n$  следующих строк указано время  $h, m, s$ , которое показывают  $i$ -ые часы. Целые числа  $h, m$  и  $s$  указывают количество часов, минут и секунд соответственно ( $0 \leq h < 12, 0 \leq m < 10^6, 0 \leq s < 10^6$ ).

### Output

В единственной строке запишите через пробел три целых числа  $h, m$  и  $s$  — наименьшее суммарное время перевода, где  $h, m$  и  $s$  — целые числа, указывающие количество часов, минут и секунд соответственно ( $0 \leq h < 12, 0 \leq m < 10^6, 0 \leq s < 10^6$ ).

### Examples

stdin	stdout
2 10 0 0 3 0 0	5 0 0
3 11 999999 999999 0 0 0 11 999999 999999	0 0 2

## Problem H. Обед

Input file: `stdin`  
Output file: `stdout`  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

Болото имеет вид узкой полосы длины  $n$ . Поверхность болота сплошь покрыта листьями водяной лилии размера 1, пронумерованными от 1 до  $n$ , на каждом из которых расположилась муха. На одном из листьев вместо мухи сидит лягушонок Квайт, и у него начинается обед. Он может прыгнуть на соседний лист или перепрыгнуть через один лист в любую сторону. Приземлившись, он съест сидящую там муху. Квайт уже большой лягушонок, а листья не очень надежные, поэтому, когда он прыгает, лист, на котором он находился, начинает тонуть.

Для того чтобы насытиться, Квайту необходимо съесть всех мух, начав свой путь с листа под номером  $s$  и закончив на листе под номером  $f$ . При прыжке на соседний лист Квайт тратит больше энергии, чем при прыжке через лист.

Необходимо спланировать прыжки лягушонка так, чтобы он смог насытиться, затратив минимальное количество прыжков на соседние листья.

### Input

В единственной строке записаны три целых числа  $n, s, f$  ( $2 \leq n \leq 10\,000, 1 \leq s, f \leq n$ ) — количество листьев на болоте, номер стартового и номер финишного листьев.

### Output

Выведите минимальное количество прыжков на соседние листья, которое потребуется для насыщения лягушонка. Если насытиться не получится, выведите единственное число  $-1$ .

### Examples

<code>stdin</code>	<code>stdout</code>
4 1 2	1

## Problem I. Бухгалтерская система счисления

Input file:            stdin  
Output file:           stdout  
Time limit:           5 секунд  
Memory limit:        256 мегабайт

Новейшая Бухгалтерская Система Счисления — самая бухгалтерская во всём мире. Её автор Цейзенпок — лучший специалист соответствующего министерства. Любое целое положительное число  $n$  в этой системе по основанию  $m$  представляется в виде суммы  $m$  слагаемых:

$$n = C_{x_m}^m + C_{x_{m-1}}^{m-1} + C_{x_{m-2}}^{m-2} + \dots + C_{x_1}^1,$$

причем  $x_1, x_2, \dots, x_m$  — целые числа такие, что  $0 \leq x_1 < x_2 < \dots < x_m$ . Числа  $C_k^m = \frac{k!}{m!(k-m)!}$  наши специалисты называют бухгалтерскими коэффициентами. Каждое число  $n$  в этой системе записывается в виде  $n = \overline{(x_m) \dots (x_2)(x_1)}$ , причём считается, что  $0! = 1$  и  $C_k^m = 0$ , если  $m > k$ . Например, число 9 в бухгалтерской системе по основанию 3 записывается в виде **(4)(3)(2)**, так как  $9 = C_4^3 + C_3^2 + C_2^1$ , а число 1 в этой системе по основанию 2 выглядит так: **(2)(0)**, поскольку  $1 = C_2^2 + C_0^1$ .

Вам необходимо составить программу, которая находит представление целого положительного числа  $n$  в бухгалтерской системе счисления по основанию  $m$ .

### Input

В единственной строке записаны два целых числа  $n$  и  $m$  ( $1 \leq n \leq 10^{16}$ ,  $2 \leq m \leq 1000$ ).

### Output

Единственная строка должна содержать последовательность из  $m$  разделенных пробелом целых чисел  $x_m, \dots, x_2, x_1$ , образующих запись числа  $n$  в бухгалтерской системе счисления. Число  $x_m$  является первой (слева) цифрой в записи числа  $n$ , а  $x_1$  — его последней цифрой.

### Examples

stdin	stdout
9 3	4 3 2
5 2	3 2

## Problem J. Формула Цейзенпока

Input file: `stdin`  
Output file: `stdout`  
Time limit: 1 секунд  
Memory limit: 256 мегабайт

Ученый Цейзенпок с планеты `i1c5l` стал известен на всю Вселенную благодаря своему недавнему открытию — формуле Цейзенпока. У этой формулы всего три аргумента:  $n$ ,  $k$  и  $m$ , а значением этой формулы является число сочетаний из  $n$  по  $k$  по модулю  $m$ .

Пока вся Вселенная думает, для каких целей применима данная формула, нам требуется автоматизировать её вычисление.

### Input

Входной файл содержит три целых числа  $n, k, m$ , разделенных пробелами ( $1 \leq n \leq 10^{18}$ ,  $0 \leq k \leq n$  и  $2 \leq m \leq 1\,000\,000$ ).

### Output

Выведите значение формулы для данных  $n, k, m$ .

### Examples

<code>stdin</code>	<code>stdout</code>
2 1 3	2
4 2 5	1

## Problem K. Мы делили апельсин

Input file:        **stdin**  
Output file:       **stdout**  
Time limit:        1 секунда  
Memory limit:     256 мегабайт

На далекой планете `ilc51` обществом из  $n$  человек собран урожай в  $k$  апельсинов. Теперь стоит задача дележа урожая.

В этом обществе принята цветовая дифференциация штанов с элементами демократии, поэтому дележ урожая происходит следующим образом. В зависимости от цвета штанов каждый человек получает ранг от 1 до  $n$ . У всех людей штаны разного цвета. Далее человек с рангом 1 объявляет свое решение: кому и сколько апельсинов достанется. После этого все  $n$  человек голосуют «за» или «против». Если хотя бы половина людей проголосует «за», то решение объявляется принятым, в противном случае, человека, внесшего предложение, изгоняют из общества, а право объявить предложение переходит к человеку с рангом 2, и процедура повторяется.

При принятии решений и голосовании каждый человек действует оптимально для себя: пытается получить как можно больше апельсинов и при вариантах с равным количеством апельсинов стремится, чтобы в обществе осталось как можно меньше людей. Если человека изгоняют из общества, то считается, что он получил отрицательное число апельсинов. Из всех оптимальных решений человек может выбрать любое. Каждый человек знает, что остальные тоже действуют оптимально, руководствуясь теми же принципами.

Но один из этих людей обладает уникальными штанами-хамелеонами. Эти штаны могут по желанию хозяина принимать один из  $m$  цветов. Никакой из этих цветов не встречается у остальных людей.

Обладатель хамелеоновых штанов точно знает, какой ранг он получит для каждого цвета своих штанов. Ему осталось выяснить минимальное и максимальное количество апельсинов, которое он получит при каждом из цветов штанов.

### Input

В первой строке целые числа  $n$ ,  $k$  и  $m$  ( $1 \leq n, k \leq 10^9$ ,  $1 \leq m \leq 10^5$ ) — количество людей, апельсинов и возможных цветов штанов-хамелеонов соответственно.

Во второй строке  $m$  целых чисел  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq n$ ), где  $a_i$  — ранг обладателя хамелеоновых штанов при цвете  $i$ .

### Output

Для каждого  $a_i$  на отдельной строке вывести минимальное и максимальное количество апельсинов, которые получит обладатель штанов цвета  $i$  или «-1 -1» (без кавычек), если его изгонят из общества.

### Examples

stdin	stdout
2 10 2	10 10
1 2	0 0
3 1 2	0 0
1 3	1 1

### Note

В первом примере человек первого ранга забирает все апельсины себе и голосует «за».

Во втором примере человек с рангом 1 должен отдать апельсин человеку третьего ранга. Если первый заберет апельсин себе, то остальные проголосуют "против". Если первый отдаст апельсин второму, то второй все равно будет «против», т.к. знает, что, изгнав первого, он тоже получит апельсин, но людей в обществе останется меньше. Так как третий тоже будет «против», то этот вариант не подходит для первого.

## Problem L. Бассейн счастья

Input file:            стандартный ввод  
Output file:          стандартный вывод  
Time limit:            1 секунда  
Memory limit:        256 мегабайт

В Казани к предстоящему Чемпионату мира по водным видам спорта построен бассейн на  $N$  дорожек. По некоторым из этих дорожек уже плавают люди.

Наблюдая за людьми, татарские учёные разделили дорожки на счастливые и несчастливые. Несчастливой дорожка считается в том случае, если по ней плавает максимальное количество людей. То есть, нет другой дорожки, по которой плавало бы больше людей, чем по этой. По несчастливым дорожкам плавают несчастные люди. Все остальные дорожки считаются счастливыми. По счастливым дорожкам плавают счастливые люди.

Учёные решили сделать как можно больше людей счастливыми. Для этого они согласовали с администрацией бассейна возможность переместить одного человека на одну из *соседних* дорожек, если это потребуется. При этом с первой дорожки пловца можно переместить только на вторую, а с последней только на предпоследнюю.

### Input

В первой строке записано единственное целое число  $N$  — количество дорожек в бассейне ( $3 \leq N \leq 10^5$ ). Во второй строке  $N$  целых чисел  $p_i$ , разделённых пробелом — распределение людей по дорожкам, где число  $p_i$  задаёт количество пловцов на  $i$ -ой дорожке ( $0 \leq p_i \leq 10^5$ ).

### Output

Выведите единственное число — минимально возможное число несчастных пловцов.

### Examples

стандартный ввод	стандартный вывод
3 1 3 5	5
4 1 0 1 0	2

## Problem M. #TheDress

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

После высадки на планете `i1c5l` люди заметили, что среди аборигенов очень популярна сине-черная одежда. У каждого аборигена в гардеробе есть хотя бы один сине-черный предмет. Этот факт мог бы и не заинтересовать людей, если бы не одно «но» — аборигены все как один утверждали, что это не сине-черная одежда, а бело-золотая.

Таким образом появился простой тест, однозначно отличающий инопланетянина от человека. На одной из межвидовой свадеб аборигена и симпатичной землянки люди сделали фотографию сине-черного платья матери жениха. Эту фотографию люди показывали тестируемому и просили назвать цвет платья. Если в ответе содержалось «blue» и «black», то это однозначно говорило о том, что тестируемый — человек с Земли. Если в ответе содержалось «white» и «gold», то это указывало на существо с планеты `i1c5l`. Если же ответ не удовлетворял ни одному из условий, то тестируемый был залётным пришельцем с другой планеты.

Перед вами полная история опросов живых существ, проведённого на планете `i1c5l`. Ваша задача — определить состав населения планеты на основе проведённого опроса.

### Input

В первой строке записано единственное целое число  $N$  — количество опрошенных существ ( $1 \leq N \leq 100$ ). Следующие  $N$  строк содержат ответы опрошенных. Каждая строка не пуста, имеет длину не более 100 символов и задаёт ответ одного из опрошенных. Ответ состоит из строчных букв латинского алфавита и пробелов. Гарантируется, что ни один ответ не содержит одновременно «blue», «black», «white» и «gold».

### Output

Выведите три числа, описывающих население планеты, каждое на отдельной строке.

Первое число — количество землян, выраженное в процентах по отношению к общему количеству опрошенных.

Второе число — количество аборигенов с планеты `i1c5l`, выраженное в процентах по отношению к общему количеству опрошенных.

Третье число — количество залётных пришельцев, выраженное в процентах по отношению к общему количеству опрошенных.

Все числа выведите с точностью не менее  $10^{-5}$ .

### Examples

стандартный ввод	стандартный вывод
3 goldandwhite white and pinkman blueblueblue and a little bit black	33.3333333333 33.3333333333 33.3333333333
4 this dress is blue and black this dress is goldblackblue eto plate kazhetsia sirenevenkoe no comments	50.0000000000 0.0000000000 50.0000000000

## Problem N. Жаба в пустыне

Input file:        stdin  
Output file:        stdout  
Time limit:        1 секунда  
Memory limit:      256 мегабайт

В точке  $A$  бескрайней пустыни закопана стратегическая атомная подводная лодка «Жаба». Командование приняло решение сменить место базирования АПЛ на точку  $B$ . С командованием спорить бессмысленно, поэтому встал вопрос как переместить закопанную подводную лодку из одной точки плоской пустыни в другую.

Выяснилось, что проектировщики АПЛ продумали этот вопрос еще на этапе строительства и заложили в её конструкцию самый настоящий телепорт. Казалось, что проблема решена, но выяснилось что телепорт имеет всего  $K$  режимов работы, которые отличаются между собой только расстоянием перемещения объекта, то есть для каждого режима известно расстояние  $L_i$ , на которое переместится лодка в заданном направлении при активации телепорта.

Необходимо найти минимальный по суммарной длине маршрут перемещения АПЛ с использованием телепорта.

### Input

В первой строке пять целых чисел  $X_A, Y_A, X_B, Y_B$  ( $-150 \leq X_A, Y_A, X_B, Y_B \leq 150$ ) — декартовы координаты двух различных точек  $A$  и  $B$ , и  $K$  ( $0 < K \leq 5$ ) — количество режимов работы телепорта.

Во второй строке  $K$  целых чисел  $L_i$  ( $0 < L_i \leq 150$ ) — расстояние телепортации в каждом из режимов.

### Output

В первой строке необходимо вывести длину минимального маршрута.

Во второй строке вывести число  $M$  — количество телепортаций в минимальном по длине маршруте.

В последующих  $M$  строках выведите по 2 действительных числа с точностью до  $10^{-6}$  — координаты  $(x_i, y_i)$ , в которых АПЛ окажется после  $i$ -ой телепортации.

Если минимальных по длине маршрутов несколько, выведите любой из них.

### Examples

stdin	stdout
1 1 4 5 3 2 3 5	5 2 2.2000000000 2.6000000000 4.0000000000 5.0000000000
0 0 18 0 2 10 5	20 3 4 3 14 3 18 0



## Problem O. Скряги.net

Input file:            **stdin**  
Output file:           **stdout**  
Time limit:            1 секунда  
Memory limit:         256 мегабайт

В деревне Скрягино наконец-то появился Интернет. Ну как появился... По территории деревни прошла абсолютно прямая кабельная трасса в районный центр. Долго гордившись наличием Интернета в деревне, жители, наконец, решили подключиться к кабелю. Эксплуатирующая организация решила организовать только одну точку подключения, зато позволила деревенским самостоятельно выбрать место подключения.

Разумеется, скрягинцы хотят провести сеть во все  $k$  домов и максимально сэкономить на стоимости проводов. Было решено, что проводка будет осуществляться от каждого дома до точки подключения по прямой, невзирая на трудности монтажа.

Располагая данными о координатах всех домов, необходимо найти точку на кабеле, суммарное расстояние от которой до всех домов будет минимальным.

### Input

В первой строке записаны две пары целых чисел  $X_1, Y_1, X_2, Y_2$  ( $-100 \leq X_1, Y_1, X_2, Y_2 \leq 100$ ) — декартовы координаты двух различных точек, лежащих на прямой магистрального кабеля. Во второй строке целое число  $k$  ( $1 \leq k \leq 10\,000$ ) — количество домов. В последующих  $k$  строках по два целых числа  $X_i$  и  $Y_i$  ( $-10\,000 \leq X_i, Y_i \leq 10\,000$ ) — координаты домов.

### Output

Выведите суммарную длину проводки и координаты искомой точки. Если таких точек несколько, выведите любую. Ответ считается верным, если абсолютная или относительная погрешность не превосходит  $10^{-5}$ .

### Examples

stdin	stdout
2 0 6 0	3.0000000000
1	2.0000000000 0.0000000000
2 3	

## Problem P. Суперфакториальная система счисления

Input file:        stdin  
Output file:       stdout  
Time limit:        1 секунда  
Memory limit:     256 мегабайт

*Эта весёлая планета...*

---

На самой совершенной из планет `11c51` всё трезво, всё разумно, всё толково... Чего там только не было, чего там только нет... Например, на соревнованиях по программированию жители планеты применяют разные системы счисления в разных дивизионах. Во втором дивизионе они используют *суперфакториальную систему счисления*. В этой системе любое положительное рациональное число представляется в виде линейной комбинации чисел, обратных к факториалам:

$$\frac{p}{q} = a_1 + \frac{a_2}{2!} + \frac{a_3}{3!} + \dots + \frac{a_n}{n!}.$$

Здесь  $a_1$  — целое неотрицательное число, а все остальные целые  $a_k$  удовлетворяют неравенствам  $0 \leq a_k < k$ . Незначащие нули в конце суперфакториальной записи числа  $\frac{p}{q}$  отбрасываются.

Необходимо найти представление положительного рационального числа  $\frac{p}{q}$  в суперфакториальной системе счисления.

### Input

В единственной строке записаны через пробел два целых числа  $p$  и  $q$  ( $1 \leq p \leq 10^6$ ,  $1 \leq q \leq 10^6$ ).

### Output

Единственная строка должна содержать последовательность разделенных пробелом целых чисел  $a_1, a_2, \dots, a_n$ , образующих запись числа  $\frac{p}{q}$  в суперфакториальной системе счисления. Если таких представлений несколько, выведите любое из них.

### Examples

stdin	stdout
1 2	0 1
2 10	0 0 1 0 4
10 2	5

## Problem Q. Микросхемы

Input file:            **stdin**  
Output file:           **stdout**  
Time limit:            1 секунда  
Memory limit:         256 мегабайт

— Я спросил электрика Петрова...  
Детская народная поэзия.

---

Наверное, вам известно, как читать микросхемы. Прежде всего, необходимо обратить внимание на особенности соединения контактов. На схемах контакты соединяются токопроводящими жилами. Если две жилы не пересекаются, то соединение между контактами, из которых выходят эти жилы, отсутствует.

У вас в руках микросхема в виде круга, на границе которого расположено  $n$  контактов. Вам необходимо подсчитать количество способов нанести ровно  $k$  непересекающихся жил, каждая из которых соединяет ровно два контакта.

### Input

В единственной строке записаны два целых числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 40$ ).

### Output

Выведите искомое количество способов.

### Examples

stdin	stdout
4 2	2
4 3	0