

Задача А. Максимальное передаточное число в системе шестерен

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

У инженера Васи есть экспериментальная установка, состоящая из n шестеренок, выстроенных в прямую линию. Каждая шестеренка характеризуется своим количеством зубьев.

Передаточное число между двумя соседними шестеренками вычисляется как отношение количества зубьев ведущей шестеренки к количеству зубьев ведомой. Для системы из нескольких шестеренок, передаточное число вычисляется как произведение передаточных чисел между соседними шестеренками.

Помогите Васе найти подотрезок подряд идущих шестеренок, который дает максимальное передаточное число на выходе, если вращаться будет левая шестеренка в подотрезке.

Формат входных данных

В первой строке содержится целое число n ($1 < n \leq 10^5$) — количество шестеренок.

Во второй строке содержится массив a из n целых чисел ($1 \leq a_i \leq 10^9$) — количество зубьев каждой шестеренки.

Формат выходных данных

Максимальное передаточное число, которое можно получить на любом подотрезке системы, выраженное в виде $\frac{P}{Q}$. Выведите два числа через пробел P и Q .

Система оценки

№	Ограничения	Баллы
1	$n \leq 100$	20
2	$n \leq 2000$	20
3	$n \leq 10^5; a_i \leq 100$	30
4	$n \leq 10^5$	30

Пример

стандартный ввод	стандартный вывод
5 6 3 8 4 2	4 1

Задача В. Улучшение системы шестеренок

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

У инженера Васи есть экспериментальная установка, состоящая из n шестеренок, выстроенных в прямую линию. Каждая шестеренка характеризуется своим количеством зубьев. Передаточное число между двумя соседними шестеренками вычисляется как отношение количества зубьев ведущей шестеренки к количеству зубьев ведомой. Для системы из нескольких шестеренок, передаточное число вычисляется как произведение передаточных чисел между соседними шестеренками.

Вася обнаружил, что общее передаточное число от первой до последней шестеренки зависит только от количества зубьев на первой и последней шестеренках. Помогите Васе найти минимальное количество операций обмена двух шестеренок, необходимое для достижения максимального общего передаточного числа системы.

Формат входных данных

В первой строке содержится целое число n ($1 < n \leq 10^5$) — количество шестеренок. Во второй строке содержится массив a из n целых чисел ($1 \leq a_i \leq 10^9$) — количество зубьев каждой шестеренки в исходном порядке.

Формат выходных данных

В первой строке выведите целое число k — минимальное количество операций обмена. В каждой из следующих k строк выведите по два целых числа i и j ($1 \leq i, j \leq n$) — номера позиций шестеренок, которые нужно поменять местами.

Система оценки

№	Ограничения	Баллы
1	$n \leq 10$	20
2	$n \leq 1000$	20
3	$n \leq 10^5$; все a_i различны	30
4	$n \leq 10^5$	30

Примеры

стандартный ввод	стандартный вывод
2 974365976 715806713	0
3 3 1 2	1 2 3
3 2 3 1	1 1 2

Задача С. Встреча в плацкартном поезде

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

Камиль и Вадим зашли в плацкартный поезд с разных концов. Поезд состоит из n вагонов, пронумерованных от 1 до n . Для каждого вагона известна его *степень заполненности* a_i — целое число (чем больше людей в вагоне, тем сложнее через него пройти).

В поезде имеется $n + 1$ *тамбур*:

- тамбур 0 — перед первым вагоном;
- тамбур i — между вагонами i и $i + 1$ (для $1 \leq i \leq n - 1$);
- тамбур n — после последнего вагона.

Камиль заходит в поезд через тамбур 0 и идёт вправо. Вадим заходит через тамбур n и идёт влево. Они хотят встретиться в одном и том же тамбуре (с одинаковым номером).

Сложность пути — это сумма значений a_i всех вагонов, через которые необходимо пройти, чтобы добраться от исходного тамбура до выбранного.

- Чтобы дойти от тамбура 0 до тамбура k ($1 \leq k \leq n$), нужно пройти через вагоны $1, 2, \dots, k$.
- Чтобы дойти от тамбура n до тамбура k ($0 \leq k \leq n - 1$), нужно пройти через вагоны $n, n - 1, \dots, k + 1$.

Требуется выбрать номер тамбура, в котором абсолютная разность сложностей путей Камиля и Вадима будет минимальной.

Если существует несколько таких тамбуров, необходимо выбрать тамбур с наименьшим номером.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — количество вагонов.

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 1000$) — степени заполненности вагонов.

Формат выходных данных

Выведите одно целое число — номер тамбура от 0 до n , в котором Камиль и Вадим должны встретиться.

Если несколько тамбуров дают одинаковую минимальную разность сложностей, выведите тамбур с наименьшим номером.

Система оценки

№	Дополнительные ограничения	Баллы
1	$n \leq 1000$	40
2	все значения a_i одинаковы	20
3	без дополнительных ограничений	40

Примеры

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

Замечание

В первом примере суммарная заполненность всех вагонов равна 15.

- Тамбур 0: путь Камиля имеет сложность 0, путь Вадима — 15, разность 15.
- Тамбур 1: Камиль проходит вагон 1 (1), Вадим — вагоны 2...5 (14), разность 13.
- Тамбур 2: Камиль проходит вагоны 1 и 2 (3), Вадим — вагоны 3...5 (12), разность 9.
- Тамбур 3: Камиль проходит вагоны 1...3 (6), Вадим — вагоны 4 и 5 (9), разность 3.
- Тамбур 4: Камиль проходит вагоны 1...4 (10), Вадим — вагон 5 (5), разность 5.
- Тамбур 5: путь Камиля — 15, путь Вадима — 0, разность 15.

Задача D. Оптимальная поездка

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

Петя едет на машине по прямой дороге длиной S километров. В начале пути его скорость равна a километров в час. В любой момент времени Петя может выбрать текущую скорость v , причём она всегда должна удовлетворять

$$a \leq v \leq a + d.$$

Энергия, которую он тратит на движение, зависит от выбранной скорости. Известно, что на любом участке пути длиной L , который Петя проезжает с постоянной скоростью v , он тратит энергию

$$E_{\text{seg}} = k \cdot v \cdot L,$$

где k — заданный коэффициент, а v — скорость на этом участке.

Суммарная энергия, которую Петя может потратить за всю дорогу, ограничена сверху значением E . Гарантируется, что энергии всегда достаточно, чтобы проехать весь путь с минимально возможной скоростью: выполняется неравенство

$$E \geq k \cdot a \cdot S.$$

Петя хочет проехать весь путь длиной S километров за *минимально возможное время*. Среди всех способов, дающих одно и то же минимальное время, его интересует тот, при котором суммарная затраченная энергия *минимальна*.

Требуется найти:

- время T , которое Петя потратит на поездку;
- энергию En , которую он при этом израсходует.

Ответ считается правильным, если абсолютная или относительная погрешность каждого из чисел T и En не превосходит 10^{-6} .

Формат входных данных

Первая строка содержит пять целых чисел:

$$a, d, S, k, E$$

$(1 \leq a \leq 10^6, 0 \leq d \leq 10^6, 1 \leq S \leq 10^6, 1 \leq k \leq 10^6, 1 \leq E \leq 10^{18})$.

Также гарантируется, что $E \geq k \cdot a \cdot S$ и $k \cdot (a + d) \cdot S \leq 10^{18}$.

Формат выходных данных

Выведите два вещественных числа:

- T — время (в часах), затраченное Петей на поездку;
- En — суммарную энергию, которую он потратит.

Ваш ответ будет засчитан, если абсолютная или относительная погрешность каждого из чисел не превосходит 10^{-6} .

Система оценки

Во всех подзадачах выполняются ограничения из условия, включая $E \geq k \cdot a \cdot S$.

№	Дополнительные ограничения	Баллы
1	$d = 0$	30
2	$E \geq k \cdot (a + d) \cdot S$	30
3	без дополнительных ограничений	40

Примеры

стандартный ввод	стандартный вывод
10 10 100 1 2000	5.0000000000 2000.0000000000
10 20 100 1 1500	6.6666666667 1500.0000000000

Задача Е. Оптимальные рейсы лифта

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

Сергею поручили оптимизировать работу лифта в высотном бизнес-центре. В здании есть один лифт и несколько сотрудников, которые хотят спуститься на первый этаж после работы.

Для каждого теста задано количество сотрудников m , ограничение на число остановок за один рейс лифта k и этажи, на которых находятся сотрудники $a_1, a_2, \dots, a_m$.

Правила работы лифта:

- Лифт всегда начинает рейс на **первом этаже** и заканчивает на **первом этаже**.
- За один рейс лифт может сделать не более k остановок (не считая первого этажа).
- На одном этаже может находиться несколько сотрудников — лифт забирает всех за одну остановку.
- За один рейс лифт может остановиться на любых этажах в **произвольном порядке**, но общее количество остановок в этом рейсе не должно превышать k .
- Важно, чтобы в итоге **все сотрудники** оказались на первом этаже, то есть каждый этаж, на котором кто-то находится, должен быть посещён хотя бы в одном из рейсов.

Задача игрока состоит в том, чтобы составить план рейсов лифта так, чтобы общее количество рейсов было как можно **меньше**.

Маршрут описывается следующим образом: для каждого рейса указывается, на скольких этажах лифт остановится и список этих этажей.

Формат описания одного теста (файла `inputX.txt`):

- Первая строка: два целых числа m и k — количество сотрудников и максимальное число остановок за один рейс.
- Вторая строка: m целых чисел $a_1, a_2, \dots, a_m$ — этажи, на которых находятся сотрудники.

Гарантируется, что $1 \leq m \leq 20$, $1 \leq k \leq 100$, $2 \leq a_i \leq 10^9$.

Формат описания ответа для одного теста:

- Первая строка: одно целое число r — количество рейсов, которые выполнит лифт.
- Далее следует r строк, описывающих каждый рейс:
 - в начале строки записано число t_i — количество остановок в i -м рейсе ($1 \leq t_i \leq k$);
 - далее записаны t_i целых чисел $x_{i,1}, x_{i,2}, \dots, x_{i,t_i}$ — номера этажей, на которых остановится лифт в этом рейсе.

Этажи в одном рейсе можно указывать в любом порядке. Важно, чтобы каждый этаж, на котором находится хотя бы один сотрудник, встречался хотя бы в одном рейсе.

Если вы не знаете решение для теста, можете вывести единственное число 0 — в этом случае за данный тест вы получите 0 баллов.

Даны четыре тестовых файла, для которых требуется найти план рейсов: `input1.txt`, `input2.txt`, `input3.txt`, `input4.txt`. Если вы не знаете ответ для какого-то теста, выведите для него 0.

Как ответ на эту задачу прикрепите **txt**-файл.

- Если вы отправляете **txt**-файл, то в нём должны подряд описываться ответы для тестов 1, 2, 3 и 4 **в указанном порядке**. Для каждого теста:
 - сначала строка с числом r — количеством рейсов,
 - затем r строк с описанием рейсов в формате, указанном выше.

Между ответами для разных тестов можно оставлять пустые строки.

Тесты

Ниже приведено содержимое четырёх файлов `inputX.txt`.

Тест 1 (`input1.txt`):

```
6 2
9 10 2 3 5 8
```

Тест 2 (`input2.txt`):

```
10 3
2 2 5 5 8 8 10 10 7 3
```

Тест 3 (`input3.txt`):

```
15 4
2 4 6 8 10 12 14 16 18 20 3 5 7 9 11
```

Тест 4 (`input4.txt`):

```
20 3
100 50 75 60 90 80 110 120 55 65 95 85 130 140 150 160 170 180 190 200
```

Каждый тест оценивается от 0 до 25 баллов по формуле

$$25 \cdot \frac{r_{\text{best}}}{r_{\text{participant}}},$$

где $r_{\text{participant}}$ — количество рейсов в вашем плане для данного теста, r_{best} — минимальное количество рейсов в эталонном решении жюри. Значение дроби не превышает 1, так как эталонное решение является оптимальным.

Если ответ для теста некорректен (формат нарушен, есть рейс с $t_i > k$, указаны этажи вне диапазона, либо не все нужные этажи покрыты хотя бы одним рейсом), то за этот тест выставляется 0 баллов.

Задача F. Испытание дрона

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

Инженерная лаборатория проводит испытания автономного навигационного дрона в двумерной среде. Среда характеризуется векторным полем, которое изменяется во времени и влияет на движение дрона. На каждом шаге дрон может выбирать собственное направление и скорость, но его возможности ограничены.

Пусть (v_x, v_y) — выбранная скорость дрона относительно среды, а (q_x, q_y) — вектор поля в данный момент времени. Тогда фактическое смещение дрона за одну секунду равно $(v_x + q_x, v_y + q_y)$.

Дрон стартует из точки (S_x, S_y) и должен попасть в точку (E_x, E_y) ровно через k секунд. Во время эксперимента векторное поле изменяется n раз. Каждый из этих интервалов характеризуется длительностью t_i секунд и постоянным значением поля (q_{x_i}, q_{y_i}) . Сумма всех длительностей равна общему времени испытания:

$$t_1 + t_2 + \dots + t_n = k.$$

Необходимо определить, существует ли стратегия управления, позволяющая достичь цели за ровно k секунд. Если это возможно, требуется вывести допустимую траекторию дрона — его координаты после каждой секунды.

Формат входных данных

Первая строка содержит четыре целых числа: S_x, S_y, E_x, E_y ($-10^4 \leq S_x, S_y, E_x, E_y \leq 10^4$) — координаты начальной и конечной точек.

Вторая строка содержит три целых числа: n, k, v_{max} ($1 \leq n, k, v_{max} \leq 10^5$) — количество интервалов, общее время полёта и максимальная собственная скорость дрона.

Каждая из следующих n строк содержит три целых числа: t_i, q_{x_i}, q_{y_i} — длительность t_i секунд, в течение которых вектор поля равен (q_{x_i}, q_{y_i}) .

Гарантируется, что:

$$t_1 + t_2 + \dots + t_n = k, \quad \sqrt{q_{x_i}^2 + q_{y_i}^2} \leq v_{max}.$$

Формат выходных данных

Выведите строку «Yes», если дрон может достичь цели ровно за k секунд, или «No» — если это невозможно.

Если ответ «Yes», выведите k строк, i -я из которых должна содержать два вещественных числа x_i, y_i — координаты дрона после i -й секунды движения.

План считается корректным, если:

- для каждого шага можно выбрать скорость (v_x, v_y) с $|v_x^2 + v_y^2|^{1/2} \leq v_{max}$, чтобы после учёта поля отклонение от указанных координат не превышало 10^{-5} ;
- расстояние между конечной точкой (x_k, y_k) и целью (E_x, E_y) не больше 10^{-5} .

Система оценки

Подзадача	Ограничения	Баллы
1	$f_{x_i} = f_{y_i} = 0$ для всех i (поле отсутствует)	15
2	$n = 1$ (поле постоянно во времени)	20
3	$n \leq 1000, k \leq 1000$	25
4	без дополнительных ограничений	40

Пример

стандартный ввод	стандартный вывод
1 1 7 4	Yes
2 3 13	4.3333333333 2.0000000000
1 3 2	5.6666666667 3.0000000000
2 1 2	7.0000000000 4.0000000000

Задача G. Робот и доставка

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

Инженеры компании **MechaPost** тестируют автономного курьерского робота, который должен доставить посылку через городскую сетку. Город состоит из L строк и C столбцов клеток. Каждая клетка — это район, для передвижения через который требуется определённое количество энергии.

Если значение в клетке равно -1 , этот район недоступен (например, разрушен или закрыт на ремонт), и робот не может там проезжать. В остальных клетках указано целое число $p_{i,j}$ — базовая стоимость передвижения через этот район.

Робот стартует в точке A (строка L_1 , столбец C_1) и должен добраться до пункта назначения B (строка L_2 , столбец C_2). Перед началом пути он получает посылку массой W килограммов. Чем тяжелее посылка, тем больше энергии он тратит: переезд через район со сложностью p требует $p \cdot W$ единиц энергии.

Робот может двигаться только по соседним клеткам — вверх, вниз, влево и вправо — и не выходит за границы карты. Энергия тратится **при выезде из текущего района**; прибытие в конечную точку B энергии не требует.

Запас аккумулятора ограничен — он не может израсходовать более E единиц энергии. Нужно определить, **максимальную массу посылки W** , которую робот способен доставить из A в B , не превышая лимит энергии.

Гарантируется, что существует хотя бы один маршрут, соединяющий A и B , который не проходит через недоступные районы.

Формат входных данных

Первая строка содержит три целых числа: L , C , E ($2 \leq L, C \leq 100; 1 \leq E \leq 10^9$) — размеры карты и максимальный запас энергии.

Далее следуют L строк по C чисел $p_{i,j}$ ($1 \leq p_{i,j} \leq 10^4$ или $p_{i,j} = -1$) — энергетические стоимости передвижения по районам ($p_{i,j} = -1$ означает, что район недоступен).

Следующая строка содержит два числа: L_1 , C_1 ($1 \leq L_1 \leq L, 1 \leq C_1 \leq C$) — координаты начальной позиции робота A .

Последняя строка содержит два числа: L_2 , C_2 ($1 \leq L_2 \leq L, 1 \leq C_2 \leq C$) — координаты пункта назначения B .

Гарантируется, что $(L_1, C_1) \neq (L_2, C_2)$.

Формат выходных данных

Выведите одно целое число — максимальную массу посылки W , которую робот может доставить, не превысив доступный запас энергии E .

Система оценки

№	Дополнительные ограничения	Баллы
1	$L = C = 2$, $E \leq 100$, без недоступных районов, $A = (1, 1)$, $B = (2, 2)$	22
2	Все доступные районы имеют одинаковую сложность $p_{i,j} = k$	27
3	Между A и B существует единственный доступный путь	20
5	Без дополнительных ограничений	31

Пример

стандартный ввод	стандартный вывод
4 4 94 2 3 -1 2 1 -1 5 4 1 1 1 3 9 2 2 1 4 1 1 4	4

Задача Н. Роботы-наблюдатели

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

В исследовательском центре работает сеть из N одинаковых роботов-наблюдателей. Каждый робот измеряет некоторый параметр окружающей среды (например, уровень вибраций или шума) и отправляет на сервер свою нагрузку — неотрицательное вещественное число.

Робот должен отправлять данные каждые Z минут в течение M суток. Таким образом, в каждый момент времени

$$t = 0, Z, 2Z, \dots, (M \cdot 24 \cdot 60 - Z)$$

каждый робот должен переслать одно значение нагрузки.

Из-за помех связи часть сообщений от роботов потерялась, и на сервере возникли “дыры” в данных. По имеющимся измерениям необходимо восстановить все пропущенные значения.

Инженеры заметили, что режим работы робота в один и тот же момент суток похож от дня к дню. Поэтому они предложили восстановление по следующему правилу.

Рассмотрим робота с номером b и некоторый момент времени t (в минутах от начала наблюдений), в который значение нагрузки отсутствует.

1. Обозначим через $D = 24 \cdot 60$ количество минут в сутках.

2. Для каждого i от 1 до P рассмотрим момент времени

$$t_i = t - i \cdot D,$$

то есть тот же самый момент суток, но i -й предыдущий день. Для каждого такого t_i известна нагрузка робота b .

3. Для каждого t_i вычислим среднюю нагрузку за предыдущий час: возьмём все моменты

$$t_i - Z, t_i - 2Z, \dots, t_i - 60$$

(их ровно $60/Z$, так как 60 кратно Z), для каждого из них известна нагрузка робота b , и посчитаем среднее арифметическое этих значений.

4. Относительную нагрузку в момент t_i определим как отношение нагрузки в момент t_i к средней нагрузке за предыдущий час для этого же робота.

5. Найдём среднее значение относительной нагрузки по всем P предыдущим дням для этого времени суток.

6. Теперь, для текущего дня и момента t вычислим среднюю нагрузку за предыдущий час робота b по тем же правилам: возьмём моменты

$$t - Z, t - 2Z, \dots, t - 60$$

и найдём среднее арифметическое нагрузок.

7. Восстановленное значение нагрузки робота b в момент времени t равно произведению средней относительной нагрузки (по P предыдущим дням) на среднюю нагрузку за предыдущий час в текущий день.

Известно, что:

- во **все моменты первого часа каждого дня** (то есть для всех t от 0 до $60 - Z$ включительно для каждого дня) данные по всем роботам есть;

- в первые P дней наблюдений нет ни одной “дыры” в данных ни у одного робота;
- во входных данных присутствуют все измерения, которые удалось получить с роботов, а пропуски нужно восстановить описанным способом;
- гарантируется, что описанный метод восстановления для всех требуемых моментов корректно определён.

По данным, полученным с роботов за последние M дней, требуется восстановить все пропущенные значения нагрузок.

Формат входных данных

Первая строка содержит пять целых чисел N , M , K , P и Z ($1 \leq N \leq 100$, $1 \leq M \leq 100$, $1 \leq K \leq 100000$, $P \leq 7$, $15 \leq Z \leq 60$). Гарантируется, что 60 кратно Z .

Считается, что наблюдения ведутся с момента времени $t = 0$ (начало первых суток) до момента $t = M \cdot 24 \cdot 60$ минут, и каждый робот должен отправлять данные во все моменты

$$t = 0, Z, 2Z, \dots, M \cdot 24 \cdot 60 - Z.$$

Далее следуют K строк, каждая содержит три целых числа L , b и t :

- L ($0 < L \leq 1000$) — нагрузка робота b в момент времени t ,
- b ($1 \leq b \leq N$) — номер робота,
- t ($0 \leq t < M \cdot 24 \cdot 60$) — время в минутах от начала наблюдений.

Гарантируется, что каждое указанное во входных данных время t кратно Z и никакая пара (b, t) не встречается во входных данных более одного раза.

Формат выходных данных

В первой строке выведите целое число R — количество восстановленных значений (количество “дыр” в данных).

В следующих R строках выведите по три значения через пробел:

- восстановленную нагрузку (вещественное число),
- номер робота b ,
- момент времени t в минутах от начала наблюдений.

Строки должны быть отсортированы сначала по возрастанию времени t , а при равном времени — по возрастанию номера робота b .

Нагрузку необходимо выводить с точностью не хуже 10^{-4} (ответ засчитывается, если абсолютная или относительная погрешность не превышает 10^{-4}).

Система оценки

В задаче каждый тест оценивается отдельно.

Пример

стандартный ввод	стандартный вывод
1 2 25 1 60	23
5 1 0	3.0000 1 1500
5 1 60	3.0000 1 1560
5 1 120	3.0000 1 1620
5 1 180	3.0000 1 1680
5 1 240	3.0000 1 1740
5 1 300	3.0000 1 1800
5 1 360	3.0000 1 1860
5 1 420	3.0000 1 1920
5 1 480	3.0000 1 1980
5 1 540	3.0000 1 2040
5 1 600	3.0000 1 2100
5 1 660	3.0000 1 2160
5 1 720	3.0000 1 2220
5 1 780	3.0000 1 2280
5 1 840	3.0000 1 2340
5 1 900	3.0000 1 2400
5 1 960	3.0000 1 2460
5 1 1020	3.0000 1 2520
5 1 1080	3.0000 1 2580
5 1 1140	3.0000 1 2640
5 1 1200	3.0000 1 2700
5 1 1260	3.0000 1 2760
5 1 1320	3.0000 1 2820
5 1 1380	
3 1 1440	

Задача I. Баланс на Оси

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

Рассмотрим прямолинейную ось, проходящую через точку 0. На левом луче этой оси отмечено L целочисленных позиций: $1, 2, \dots, L$. На правом луче также отмечено L целочисленных позиций: $1, 2, \dots, L$.

Два объекта, имеющие коэффициенты A и B соответственно, должны быть размещены на этой оси. Объект с коэффициентом A размещается на левом луче в позиции x (x — целое число, $1 \leq x \leq L$). Объект с коэффициентом B размещается на правом луче в позиции y (y — целое число, $1 \leq y \leq L$).

Ось находится в равновесии, если выполняется равенство: $A \cdot x = B \cdot y$.

Определите, существуют ли такие целые позиции x и y (по одной на каждом из лучей, $1 \leq x, y \leq L$), при которых ось будет сбалансирована.

Формат входных данных

Единственная строка входных данных содержит три целых числа A , B и L ($1 \leq A, B, L \leq 10^{18}$) — коэффициенты объектов и количество позиций на каждом луче соответственно.

Формат выходных данных

Выведите YES, если искомые позиции существуют. В противном случае выведите NO.

Система оценки

Каждая группа тестов будет оцениваться только в том случае, если все обязательные группы тестов пройдены. Баллы начисляются только если все тесты группы пройдены. Примеры не оцениваются. Все тесты разделены на группы со следующими ограничениями:

Подзадача	Ограничения	Обязательные группы	Баллы
1	$A, B \leq 10^5$		20
2	$A, B \leq 10^9$	1	30
3	$A, B \leq 10^{18}$	2	50

Примеры

стандартный ввод	стандартный вывод
2 4 5	YES
10 6 4	NO

Замечание

В первом примере одно из возможных решений — выбрать $x = 4$ и $y = 2$, тогда равенство $6 \cdot 4 = 12 \cdot 2$ будет выполняться.

Задача J. Последовательность Арсения

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

Хомяк Арсений придумал последовательность дробей:

$$p_1 = \frac{1}{1+1}, \quad p_2 = \frac{1}{1+p_1}, \quad p_3 = \frac{1}{1+p_2}, \quad p_4 = \frac{1}{1+p_3}, \quad \dots$$

то есть для всех $i \geq 2$ выполняется рекуррентное соотношение

$$p_i = \frac{1}{1+p_{i-1}}.$$

Арсений плохо считает, поэтому просит вас помочь ему найти n -ю дробь p_n .

Значение p_n может быть записано в виде дроби $\frac{a}{b}$, где a и b являются взаимно простыми.

Требуется для нескольких заданных значений n найти соответствующую дробь p_n . Если вы не знаете ответ для теста, то можете записать -1.

Как ответ на эту задачу прикрепите **txt** файл. При отправке **txt** файла на первой строке должно быть записано выражение, соответствующее ответу для первого теста, на второй строке — для второго теста и т. д.

Каждый тест оценивается в 25 баллов.

По этой задаче на проверку принимается не более трёх файлов.

Ответ для каждого теста выводите либо как a и b (два числа через пробел), либо как -1.

Номер теста	Значение n	Ответ
1	3	(поле ответа)
2	6	(поле ответа)
3	10	(поле ответа)
4	15	(поле ответа)