

Задача А. Головы и хвосты

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В битве сошлись дракон и герой. У дракона h голов и t хвостов. У героя есть волшебный меч, который одним ударом отрубает дракону ровно c отростков (голов или хвостов). Когда у дракона осталось меньше c отростков, мечом ударить невозможно.

После каждого удара героя, если отрублены все головы, герой побеждает. Иначе происходит следующее:

- если этим ударом герой отрубил хотя бы одну голову, у дракона вырастает a новых голов;
- если этим ударом герой отрубил хотя бы один хвост, у дракона вырастает b новых хвостов;
- после всего этого дракон один раз кусает героя одной головой.

Герой хочет победить дракона, получив как можно меньше укусов. Удастся ли ему победить, и если да, то каково минимальное количество укусов?

Формат входных данных

В первой строке записаны пять целых чисел, разделённых пробелами: h , t , a , b и c . Все они лежат в пределах от 1 до 10 000 включительно.

Формат выходных данных

Выведите одно целое число: минимальное количество раз, которое дракон успеет укусить героя, или -1 , если герой не сможет победить дракона.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
10 20 3 4 5	3
1 1 1 1 10	-1

Система оценки

Решения в этой задаче проверяются на 22 тестах: двух примерах и 20 основных тестах. Каждый основной тест оценивается в 5 баллов. Как и в других задачах, решение должно правильно работать на примерах — иначе проверка на основных тестах не производится.

Задача В. Целые точки

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим окружность целого радиуса r с центром в $(0, 0)$. Посчитаем, сколько на ней точек с целыми координатами: например, при $r = 2$ их всего четыре, а при $r = 5$ их двенадцать.

Дано целое число n . Найдите такой радиус r от 1 до n , для которого точек с целыми координатами на окружности как можно больше.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000\,000$).

Формат выходных данных

Выведите одно целое число r : радиус окружности. Количество точек с целыми координатами на окружности с центром в $(0, 0)$ и радиусом r должно быть не меньше, чем для любого другого целого радиуса от 1 до n . Если подходящих r несколько, выведите любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	1
5	5

Система оценки

Решения в этой задаче проверяются на 22 тестах: двух примерах и 20 основных тестах. Каждый основной тест оценивается в 5 баллов. Как и в других задачах, решение должно правильно работать на примерах — иначе проверка на основных тестах не производится.

Задача С. Камень, ножницы, бумага

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Аня и Боря играют в игру «камень, ножницы, бумага». Правила обычные: бумага побеждает камень, камень побеждает ножницы, а ножницы побеждают бумагу. Если игроки показали один и тот же знак, то засчитывается ничья. Определите результат игры по знакам, показанным Аней и Борей.

Формат входных данных

В первой строке дано название показанного Аней знака на английском языке: «rock» (без кавычек) для камня, «paper» для бумаги и «scissors» для ножниц. Во второй строке в аналогичном формате задан знак, показанный Борей.

Формат выходных данных

Выведите «Anna» (без кавычек, регистр важен), если победила Аня, «Boris», если победил Боря, и «Draw», если игра закончилась вничью.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
scissors paper	Anna
rock paper	Boris

Система оценки

Тесты к этой задаче состоят из двух групп. Баллы за каждую группу ставятся только при прохождении всех тестов, подходящих под ограничения этой группы.

Во всех тестах первой группы игра не закончилась вничью. За прохождение тестов первой группы можно получить 60 баллов.

На тесты второй группы не накладывается никаких дополнительных ограничений. За прохождение тестов второй группы можно получить 40 баллов.

Задача D. Двенадцать друзей

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Двенадцать друзей попали в плен к сумасшедшему профессору. Профессор согласен отпустить их, если они пройдут испытание.

На столе лежат в ряд двенадцать карточек: сверху они все одинаковые, а на нижних сторонах написаны имена друзей, каждое на своей карточке. Профессор выбирает одного из друзей — испытуемого — и отводит к столу. После этого испытуемый может открывать карточки по одной в любом порядке и смотреть, что на них написано. Если испытуемый найдёт своё имя, открыв не более шести карточек, то все друзья будут свободны. Если же, открыв шесть карточек, испытуемый так и не увидит своё имя — ...лучше даже не думать о последствиях.

У друзей есть программируемый робот, который может незаметно подобраться к столу, посмотреть на все карточки с обеих сторон и, возможно, поменять местами какие-то две из них. К сожалению, робот никак не может передать увиденное друзьям, иначе обман раскроется.

Помогите друзьям! Напишите программу для робота и придумайте, как действовать испытуемому, чтобы друзья гарантированно оказались на свободе.

Имена у друзей всегда одни и те же, их можно найти в примере ниже.

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. Каждый запуск — интерактивный: решение получает входные данные по строкам, и иногда должно вывести команду, чтобы получить очередную строку.

При первом запуске решение действует за робота. Входные данные начинаются со строки со словом «robot». Следующая строка содержит целое число t — количество тестовых случаев ($1 \leq t \leq 1000$). Каждый тестовый случай задан на одной строке. В этой строке записаны через пробел имена друзей в том порядке, в котором карточки с этими именами лежат на столе.

В ответ на каждую строку с именами следует вывести строку, в которой через пробел записаны два имени: имена друзей на карточках, которые робот должен поменять местами. Можно вывести два одинаковых имени, если робот не должен ничего менять. Не забудьте вывести перевод строки и очистить буфер вывода — только после этого решение получит следующий тестовый случай, если он есть.

При втором запуске решение действует за испытуемого. Входные данные начинаются со строки со словом «subject». Следующая строка содержит целое число t — количество тестовых случаев, то же, что и при первом запуске. Каждый тестовый случай начинается со строки с именем испытуемого. В каждом тестовом случае заранее зафиксировано, кто станет испытуемым, а положение карточек ровно то, которое получилось после действия робота.

Чтобы открыть карточку, решение должно вывести на отдельной строке её номер в порядке слева направо — целое число от 1 до 12. Не забудьте вывести перевод строки и очистить буфер вывода. В ответ решение получит имя, написанное на этой карточке. Как только открытое имя совпадает с именем испытуемого, тестовый случай завершается (и начинается следующий, если он есть). Если же открыто уже

шесть карточек, но имя испытуемого не было найдено, остаётся лишь немедленно завершить работу программы — и получить вердикт «Wrong Answer».

Примеры

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске.

Далее показаны два запуска какого-то решения на примере.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
robot 2 Bo Ada Dale Eve Liam Fred Kai Hiro Gina Jay Ira Cleo Dale Eve Liam Kai Cleo Ira Jay Hiro Gina Bo Ada Fred	Bo Ada Cleo Cleo
subject 2 Ada Ada Liam Dale Eve Liam	1 1 2 3

Система оценки

Решения в этой задаче проверяются на 21 тесте: одном примере и 20 основных тестах. Каждый основной тест оценивается в 5 баллов. Как и в других задачах, решение должно правильно работать на примере — иначе проверка на основных тестах не производится.

В первой половине основных тестов $t = 1$. Во второй половине основных тестов $t = 1000$.

Задача Е. Номиналы монет

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У вас есть монеты нескольких положительных целочисленных номиналов. Монеты каждого номинала присутствуют у вас в неограниченном количестве. Определите, можно ли такими монетами без сдачи набрать любую целую положительную сумму.

Формат входных данных

В первой строке дано целое число n — количество различных номиналов монет ($1 \leq n \leq 5$). Во второй строке через пробел даны n чисел a_1, a_2, \dots, a_n — номиналы монет ($1 \leq a_1 < a_2 < \dots < a_n \leq 100$).

Формат выходных данных

Выведите «Yes» (без кавычек), если имеющимися номиналами можно набрать любую целую положительную сумму без сдачи, в противном случае выведите «No». Регистр не важен.

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы, всех тестов всех предыдущих групп, а также всех примеров. Обратите внимание, что решение должно правильно работать на всех примерах (даже тех из них, которые не подходят под ограничения группы) — иначе проверка на основных тестах не производится.

Во всех тестах первой группы $n = 1$. За прохождение первой группы можно получить 20 баллов.

Во всех тестах второй группы $n \leq 2$. За прохождение второй группы можно получить ещё 40 баллов.

Во всех тестах третьей группы $n \leq 5$, то есть никаких дополнительных ограничений не накладывается. За прохождение третьей группы можно получить оставшиеся 40 баллов.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 1 5 10 50 100	yes
1 2	no
2 2 4	No

Пояснения к примерам

Монетами из первого примера можно набрать любую сумму. Например, 237 можно набрать как $100 + 100 + 10 + 10 + 10 + 5 + 1 + 1$, как $100 + 50 + 50 + 10 + 10 + 5 + 5 + 5 + 1 + 1$ и ещё многими другими способами.

Во втором и третьем примерах получится набрать только чётные суммы.

Задача F. Измерение планеты

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Нат получил приказ найти и исследовать плохо изученную карликовую планету Миттерр в глубоком космосе. Несмотря на то, что ни её положение, ни её размеры пока науке неизвестны, Нат немедленно сел в свой космический корабль Ангенс и отправился на поиски. Благодаря его познаниям в гравитационных волнах он смог определить сравнительно небольшой куб, где находится планета, и близко к нему подлететь. А именно, введём в пространстве систему координат: ось \overrightarrow{OX} , или *ось абсцисс*, направлена слева направо; ось \overrightarrow{OY} , или *ось ординат*, направлена сзади вперёд; ось \overrightarrow{OZ} , или *ось аппликата*, направлена снизу вверх. В этой системе координат каждой точке соответствует тройка координат (x, y, z) , задающая в километрах положения проекций этой точки на все три оси: например, координата x растёт при движении точки вдоль оси абсцисс, убывает при движении против неё и не меняется при перемещении в пределах плоскости, параллельной двум другим осям. Координаты x, y, z мы будем также называть *абсциссой*, *ординатой*, *аппликатой* точки, соответственно.

В этой системе координат Нат находится в точке $(0, 0, 0)$, а планета полностью находится в кубе $[1; L] \times [1; L] \times [1; L]$. Кроме того, по дифракции высокочастотного излучения, исходящего от планеты, Нат понял, что планета имеет форму шара с центром в точке с целыми координатами и целым радиусом. Формально, существуют такие целые числа $1 \leq c_x, c_y, c_z, r \leq L$, что планете принадлежат те и только те точки (x, y, z) , для которых $\sqrt{(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2} \leq r$, причём у всех таких точек верно $1 \leq x, y, z \leq L$. Впрочем, определить положение и радиус пока ещё не удалось, потому что рядом с планетой нет никакой звезды, которая бы её достаточно хорошо освещала, а ультрафиолетовые датчики на Ангенсе недостаточно чувствительны.

Помогите Нату определить радиус планеты.

Протокол взаимодействия

Сначала из первой строки считайте целое число L — размер куба, содержащего планету ($3 \leq L \leq 2020$).

Вы будете давать Ангенсу последовательные приказы о перемещении. Каждый приказ является строкой из положительного количества *команд* — символов, входящих в следующий перечень:

- «L» — перемещение на километр влево (абсцисса уменьшается на единицу);
- «R» — перемещение на километр вправо (абсцисса увеличивается на единицу);
- «B» — перемещение на километр назад (ордината уменьшается на единицу);
- «F» — перемещение на километр вперёд (ордината увеличивается на единицу);
- «D» — перемещение на километр вниз (аппликата уменьшается на единицу);
- «U» — перемещение на километр вверх (аппликата увеличивается на единицу).

Символы могут быть любого регистра — как прописные, так и строчные, таким образом, есть всего 12 разрешённых символов (шесть прописных и шесть строчных). Если строка будет иметь нулевую длину или содержать неразрешённые символы, вы получите вердикт PE (Presentation Error). Не забывайте после отправки приказа переводить строку и осуществлять сброс буфера вывода (`std::cout.flush()` в C++, `fflush(stdout)` в C, `stdout.flush()` в Python и D, `flush(output)` в Pascal и Delphi, `PrintWriter.flush()` в Java, для помощи по остальным языкам обратитесь к документации), иначе, скорее всего, вы получите ошибку IL (Idleness Limit Exceeded).

Получив приказ s , Ангенс начнёт последовательно выполнять команды из этого приказа. Если выполнение i -й команды приведёт к перемещению корабля в точку, принадлежащую планете Миттерр, то все команды, начиная с i -й, выполнены не будут, и бортовой компьютер в единственной строке выведет вам число $i - 1$ — количество успешно выполненных команд. В частности, если первая же команда из s привела бы к столкновению с планетой, то корабль останется на месте, а вам будет выведено число 0. А если Ангенс смог выполнить весь приказ без посещения точек планеты, то вы получите $|s|$ — длину строки s .

По причинам, таким как срочность поручения Ната, дефицит топлива в корабле и малая оперативная память бортового компьютера, ваш набор приказов ограничен. Общее количество приказов не должно превышать 10 000. Общее количество выполненных команд не должно превышать 100 000. Суммарная длина ваших приказов (с учётом невыполненных команд) не должна превышать 1 000 000. При нарушении любого из этих условий вы получите вердикт WA (Wrong Answer).

Если вы готовы дать ответ на ваше задание, вместо очередного приказа выведите одно целое положительное число r — радиус планеты ($1 \leq r \leq L$). После этого ваша программа должна сразу завершиться. Вывод ответа на задание не считается приказом и, соответственно, не подпадает под вышеуказанные ограничения.

Числа c_x , c_y , c_z и r в каждом тесте зафиксированы заранее и не меняются по ходу решения.

Система оценки

Все тесты в этой задаче состоят из нескольких групп. Нулевая группа — это тест из условия, в котором $L = 3$, $c_x = c_y = c_z = 2$ и $r = 1$. Ваше решение обязательно должно её пройти, чтобы оно было принято на проверку.

Первая группа состоит из тестов с $L = 4$. Если вы пройдёте все тесты этой группы, вы получите 4 балла.

Вторая группа состоит из тестов с $L = 10$. Если вы пройдёте все тесты этой группы, вы получите 10 баллов.

Третья группа состоит из тестов с $11 \leq L \leq 14$. Если вы пройдёте все тесты этой группы и все тесты первой и второй групп, вы получите ещё 14 баллов.

Четвёртая группа состоит из тестов с $L = 40$, в которой планета *касается* плоскостей $x = 1$, $y = 1$, $z = 1$ — это означает, что $c_x = c_y = c_z = r + 1$. Пройдя все тесты этой группы, вы получите 10 баллов.

Пятая группа состоит из тестов с $L = 1000$, в которых $c_x = c_y = c_z$. Пройдя все тесты этой группы и все тесты четвёртой группы, вы получите ещё 15 баллов.

Шестая группа состоит из тестов с $1001 \leq L \leq 2019$, в которых существует такое целое k , для которого планета содержит точку (k, k, k) . Пройдя все тесты этой группы и все тесты четвёртой и пятой групп, вы получите ещё 20 баллов.

Седьмая группа состоит из тестов с $L = 2020$ и $r \geq 600$. Пройдя все тесты этой группы, вы получите 17 баллов.

Восьмая группа состоит из тестов с $L = 2020$ и $r \geq 240$. Пройдя все тесты этой группы и всех предыдущих групп, вы получите оставшиеся 10 баллов.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	RrrfF
5	UUUul
1	L
0	bLU
2	dLUuur
6	duuffd
0	UFDDDDDB
2	1

Пояснение к примеру

В примере Ангенс перемещался так:

- В самом начале он находился в точке $(0, 0, 0)$.
- После первого приказа он переместился в точку $(3, 2, 0)$.
- После первой команды второго приказа он переместился в точку $(3, 2, 1)$ и дальше не смог подвинуться вверх, поскольку точка $(3, 2, 2)$ принадлежит планете.
- Ангенс не мог подвинуться влево в третьем приказе, так как точка $(2, 2, 1)$ принадлежит планете, поэтому он остался в точке $(3, 2, 1)$.
- После первых двух команд четвертого приказа корабль оказался в точке $(2, 1, 1)$, откуда не смог подвинуться вверх в точку $(2, 1, 2)$.
- Выполнив все команды пятого приказа, Ангенс оказался в точке $(2, 1, 3)$.
- В результате шестого приказа Ангенс остался на месте, так как он не мог выполнить первую же команду, требующую движения вниз в точку $(2, 1, 2)$.
- Выполнив первые две команды седьмого приказа, корабль попал в точку $(2, 2, 4)$, откуда он не смог переместиться вниз в точку $(2, 2, 3)$, находящуюся в Миттере.

Обратите внимание, что в пример добавлены лишние переводы строк для лучшей читаемости, вам их выводить не нужно.

Задача G. Сумма по треугольникам

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: **7 секунд**
Ограничение по памяти: 512 мегабайт

Вам дана (не обязательно симметричная) матрица A размера $n \times n$ из неотрицательных целых чисел, при этом все её диагональные элементы равны 0. Найдите значение суммы

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \min(A_{i,j}, A_{j,k}, A_{k,i}).$$

Формат входных данных

В целях уменьшения объёма входных данных они даны в зашифрованном виде. В единственной строке входных данных даны два целых числа n и x_0 — размерность матрицы и начальное состояние генератора случайных чисел ($3 \leq n \leq 3000$, $1 \leq x_0 \leq 10^9$).

Считав эти два числа, сгенерируйте матрицу $A_{i,j}$ по следующему правилу:

1. Выполните присваивание $x := x_0$.
2. Рассмотрите элементы $A_{i,j}$ в лексикографическом порядке пар (i, j) . То есть сперва $(1, 1)$, потом $(1, 2), \dots, (1, n)$, $(2, 1), \dots, (2, n)$ и так далее до (n, n) .
3. Для каждого элемента $A_{i,j}$:
 - если $i = j$, то присвойте $A_{i,j}$ значение 0;
 - иначе сперва выполните присваивание $A_{i,j} := x \bmod 10^7$, а потом присваивание $x := (1\,664\,525 \cdot x + 1\,013\,904\,223) \bmod 2^{32}$.

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Система оценки

Чтобы ваше решение было принято на проверку, оно должно пройти тест из условия. После этого оно будет независимо протестировано на 10 основных тестах, каждый из которых оценивается в 10 баллов.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 850860854	12851754

Замечание

Матрица в примере из условия выглядит так:

$$\begin{pmatrix} 0 & 860854 & 6150685 \\ 3423064 & 0 & 1884759 \\ 3422666 & 8109217 & 0 \end{pmatrix}$$

Всего есть 27 разных способов выбрать i, j, k . Каждый способ, в котором среди этих трёх чисел есть совпадающие, даёт вклад 0 в ответ, так как минимум из трёх неотрицательных чисел, одно из которых равно 0, равен 0. Ненулевой вклад дают только следующие способы:

- $i = 1, j = 2, k = 3$;
- $i = 2, j = 3, k = 1$;
- $i = 3, j = 1, k = 2$;
- $i = 1, j = 3, k = 2$;
- $i = 3, j = 2, k = 1$;
- $i = 2, j = 1, k = 3$.

Первые три способа дают одинаковый вклад $\min(860\,854, 1\,884\,759, 3\,422\,666) = 860\,854$ в ответ. Последние три способа также дают одинаковый вклад в ответ, равный $\min(6\,150\,685, 3\,423\,064, 8\,109\,217) = 3\,423\,064$. Получается, что ответ равен $860\,854 \cdot 3 + 3\,423\,064 \cdot 3 = 12\,851\,754$.

Матрица генерируется случайно только для ускорения ввода. Все решения жюри работают для любых матриц, недиагональные элементы которых — целые неотрицательные числа от 0 до $10^7 - 1$ включительно, а диагональные элементы равны 0.

Во всех тестах жюри значения x_0 **выбирались случайно** из диапазона $[1, 10^9]$ без дополнительного изучения структуры получившейся матрицы.

Задача Н. Гладкие префиксы

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Зафиксируем некоторое число B . Натуральное число n называется *гладким*, если любой его делитель, больший единицы, имеет делитель, больший единицы и не превосходящий B . В этой задаче B имеет не более двух натуральных делителей — то есть гарантируется, что B либо равно единице, либо простое.

Натуральное число $n = \overline{a_1 \dots a_k}$ называется *сверхгладким*, если все его непустые префиксы $\overline{a_1}$, $\overline{a_1 a_2}$, \dots , $\overline{a_1 a_2 \dots a_k}$ являются гладкими числами.

Найдите наибольшее сверхгладкое число, состоящее из ненулевых цифр.

Формат входных данных

В единственной строке находится натуральное число B — предел гладкости ($1 \leq B \leq 49\,999$). Число B либо простое, либо равно единице.

Формат выходных данных

Выведите одно натуральное число — наибольшее сверхгладкое число, состоящее только из цифр 1–9. Гарантируется, что для всех $1 \leq B \leq 49\,999$ ответ существует.

Система оценки

Чтобы ваше решение было принято на проверку, оно должно пройти все тесты из условия. После этого оно будет независимо протестировано на 100 основных тестах, каждый из которых оценивается в один балл.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	1
2	16
3	648

Пояснения к примерам

При $B = 1$ единственное гладкое число — 1.

При $B = 2$ гладкие числа — это только степени двойки. Значит, сверхгладким может быть только число, все префиксы которого являются степенями двойки, таким свойством обладают только 1, 2, 4, 8, 16.

При $B = 3$ оказывается, что наибольшее сверхгладкое число равно 648: все числа $6 = 2 \cdot 3$, $64 = 2^6$, $648 = 2^3 \cdot 3^4$ являются гладкими.

Задача I. Бесконечный небоскрёб

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Главная достопримечательность города N — небоскрёб с бесконечным количеством этажей, пронумерованных целыми положительными числами $(1, 2, 3, \dots)$. Основной метод передвижения внутри небоскрёба — лифт с $n + 1$ кнопкой: i -я из них при $1 \leq i \leq n$ — это кнопка «подняться на i этажей вверх», а $(n + 1)$ -я — «опуститься вниз на p этажей», где p — простое число, большее n . Если номер этажа, на котором вы находитесь, не превосходит p , то нажатие на последнюю кнопку ничего не делает.

Вы слышали от знакомого математика, что для того, чтобы попасть с любого этажа на любой другой, достаточно двух кнопок: любой из первых n и последней. Вы решили проверить это утверждение на практике следующим образом: сперва попасть с первого этажа на второй, нажимая только на первую и последнюю кнопки, потом со второго на третий, нажимая только на вторую и последнюю кнопки, ..., наконец, попасть с n -го этажа на $(n + 1)$ -й, нажимая только на n -ю и последнюю кнопки.

Несмотря на частое использование, последняя кнопка всё ещё работает прекрасно, а вот первые n иногда плохо нажимаются. Поэтому вы решили минимизировать суммарное количество нажатий на первые n кнопок в процессе эксперимента. Чему равно искомое минимальное количество нажатий?

Можно доказать, что для каждого i от 1 до n с любого этажа можно попасть на любой другой, используя только кнопки с номерами i и $n + 1$.

Формат входных данных

В единственной строке входных данных даны два целых числа n и p ($1 \leq n \leq 10^8$, $2 \leq p \leq 999\,999\,937$, $n < p$, p — простое).

Формат выходных данных

Выведите одно целое число — минимальное возможное суммарное количество нажатий на первые n кнопок в процессе эксперимента.

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы, всех тестов всех предыдущих групп, а также всех примеров.

Во всех тестах первой группы $n \leq 10^3$ и $p \leq 9973$. За прохождение всех тестов первой группы можно получить 12 баллов.

Во всех тестах второй группы $n \leq 10^6$ и $p \leq 999\,999\,937$. За прохождение всех тестов второй группы можно получить ещё 34 балла.

Во всех тестах третьей группы $n \leq 10^8$ и $p \leq 999\,999\,937$, то есть никаких дополнительных ограничений не накладывается. За прохождение третьей группы можно получить оставшиеся 54 балла.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 2	1
3 5	6

Пояснения к примерам

В первом примере достаточно один раз нажать на первую кнопку.

Будем обозначать нажатие на i -ю кнопку значком «+», а на последнюю — значком «-». Тогда во втором примере одна из оптимальных последовательностей нажатий такая: «+» для $i = 1$, «+++» для $i = 2$, «+-» для $i = 3$. В ней всего 6 нажатий на кнопки с первой по n -ю. Можно доказать, что нажать на них меньшее число раз не получится.

Задача J. Бассейн

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Влад тренируется в бассейне в плавании в своём любимом стиле — кроле на спине. Разумеется, он всё знает об этой технике плавания и в том числе разбирается в её минусах. Один из них — пловец не видит, что перед ним происходит! В бассейне из-за этого можно неожиданно врезаться в бортик рукой или даже головой.

Мы упрощённо вместо трёхмерного бассейна будем представлять его сечение вертикальной плоскостью, параллельной дорожкам. Влад в этой модели представлен горизонтальным отрезком FH длины h : F — ступни, а H — голова. Кроме того, на этом отрезке есть точка S — плечевой сустав, расстояние SH равно s . Из S торчат руки: левая рука SL и правая рука SR . Оба отрезка имеют длину a .

В изначальном положении ступни Влада находятся на расстоянии p от правого бортика бассейна, голова направлена направо, правая рука направлена вверх, а левая — вниз. По свистку Влад начинает плыть кролем: весь отрезок движется направо с постоянной скоростью v , а в системе отсчёта, связанной с Владом, обе руки движутся равномерно по часовой стрелке так, что кончики пальцев L и R имеют скорость w — таким образом, руки Влада всегда диаметрально противоположны, и каждая из них совершает полный оборот вокруг плечевого сустава за время $\frac{2\pi a}{w}$.

Левый бортик находится на очень большом расстоянии от Влада, поэтому Влад до него точно не достанет и через некоторое время врежется в правый бортик. Мы будем считать его вертикальной прямой, бесконечной и вверх, и вниз, и Влад врежется в тот момент, когда либо точка L , либо точка R , либо точка H впервые окажется на этой прямой.

Влад привык определять, чем он врежется в бортик, экспериментальным путём. Удивите его и напишите программу, которая по всем перечисленным параметрам определит, чем он врежется: левой рукой, правой рукой или головой.

Формат входных данных

В единственной строке находится шесть положительных целых чисел h, s, a, p, v, w — высота Влада, высота его головы относительно плечевого сустава, длина рук, изначальное расстояние от ступней Влада до правого бортика, скорость Влада и скорость вращения кончиков пальцев во время кроля ($1 \leq h, s, a, p, v, w \leq 100\,000$). Гарантируется, что $s < h < p$ и $s < a$.

Формат выходных данных

Выведите одну строку:

- «Left arm», если Влад впервые врежется левой рукой;
- «Right arm», если Влад впервые врежется правой рукой;
- «Head», если Влад впервые врежется головой.

Каждая буква может быть как прописной, так и строчной.

Гарантируется, что при данных ограничениях не произойдёт одновременных столкновений; таким образом, ответ всегда единственный.

Система оценки

Чтобы ваше решение было принято на проверку, оно должно пройти все тесты из условия. После этого оно будет независимо протестировано на 100 основных тестах, каждый из которых оценивается в один балл.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1 2 4 1 4	Left arm
2 1 3 4 1 1	Right arm
3 2 3 4 1 2	Head