

Problem A. Перекрёсток судьбы

Input file:	<i>standard input</i>
Output file:	<i>standard output</i>
Time limit:	1 секунда
Memory limit:	64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Сергей уже не первый месяц пытается найти девушку своей мечты. На просторах интернета он успел перезнакомиться не с одной сотней девушек, но почти все из них оказывались неспособными поддержать беседу, хоть сколько-то выходящую за рамки обсуждения повседневных проблем. Но вот буквально пару дней назад он нашёл Её, девушку, с которой он мог переписываться часами, обсуждая буквально всё на свете, начиная с учений Аристотеля и заканчивая современными теориями в физике. И ему даже удалось договориться с Ней о свидании в небольшом уютном кафе на перекрёстке улиц Маркса и Энгельса.

И сейчас Сергей как раз направляется в сторону этого кафе, и он чертовски опаздывает, что его совсем не радует. Сергей идёт вдоль улицы Маркса, переходит её по единственному пешеходному переходу, доходит до перекрёстка с улицей Энгельса, переходит её по светофору и, наконец, заходит в кафе. Там никого. Он опоздал, Она уже ушла. Но Сергей почему-то думает не о том, как бы Её догнать. Его мысли почему-то совсем о другом, о том, насколько правильным было решение перейти дорогу на переходе. Ведь он мог дойти до перекрёстка и перейти обе улицы по светофору. И как знать, может в этом случае он бы не потерял несколько бесценных секунд в ожидании зелёного сигнала светофора и успел вовремя. Какой из этих вариантов быстрее — зависит исключительно от того, в какой именно момент времени загорится нужный сигнал светофора, и заранее предугадать этого нельзя. Но оценить вероятность вполне можно.

Сергей всегда ходит с постоянной скоростью v и очень строго соблюдает правила дорожного движения, поэтому он выходит на дорогу только когда ему горит зелёный сигнал светофора, но не жёлтый, и уж точно не красный. Начав переходить дорогу, Сергей всё с той же постоянной скоростью и без остановок её переходит, уже независимо от дальнейших сигналов светофора. Светофор на перекрёстке Маркса и Энгельса работает самым обычным образом, т.е. t_1 секунд горит зелёный для пешеходов, переходящих через улицу Энгельса и красный для тех, кто хочет перейти через Маркса, затем t_2 секунд жёлтый для переходящих Энгельса и всё ещё красный для переходящих Маркса, затем t_3 секунд зелёный для переходящих Маркса и красный для переходящих Энгельса, затем t_4 секунд жёлтый для переходящих Маркса и всё ещё красный для переходящих Энгельса, затем всё повторяется.

Input

В первой строке дано единственное целое число v — скорость Сергея ($1 \leq v \leq 10^6$). Во второй строке даны три целых числа d , w_1 и w_2 — расстояние от пешеходного перехода до перекрёстка, ширина улиц Энгельса и Маркса соответственно ($1 \leq d, w_1, w_2 \leq 10^6$). В третьей строке даны целые числа t_1 , t_2 , t_3 и t_4 , описывающие схему работы светофора ($1 \leq t_1, t_2, t_3, t_4 \leq 10^6$). Гарантируется, что жёлтый сигнал светофора горит достаточно долго, чтобы Сергей успел перейти дорогу до того, как загорится красный даже если он выйдет на перекрёсток в самый последний момент, когда ещё горит зелёный. Можно считать, что водители всегда и мгновенно уступают дорогу на пешеходном переходе. Все расстояния даны в метрах, времена — в секундах, а скорости — в метрах в секунду.

Output

Выведите два числа — вероятность того, что при равновероятном начальном состоянии светофора Сергей дошёл бы до кафе быстрее, перейдя Маркса по пешеходному переходу, и вероятность того, что он дошёл бы быстрее, перейдя обе улицы на перекрёстке. Абсолютная погрешность вашего ответа не должна превосходить 10^{-9} .

Example

standard input	standard output
1 10 1 1 4 1 4 1	0.4 0.1

Problem B. Естественный отбор

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Никите же в этом году не до развлечений. Ещё бы — ведь ему всего через полгода поступать в университет. И, в отличие от большинства своих сверстников, Никита хочет подойти к этому процессу со всей ответственностью. В частности, он уже сейчас выбирает, на какую специальность на матмехе ему стоит поступать. В последнее время их появилось столько новых и разных, что сравнить их все между собой и выбрать лучшую — нетривиальная задача. Поэтому Никита решил подойти к этой проблеме более системно. Он выбрал несколько важных для него критериев вроде того, преподают ли перл на данной специальности или сколько семестров матанализа в программе. После этого он по каждому из критериев разбил все специальности на две группы. Сделав это, он понял, что после разбиения по одному любому критерию в каждой из групп остаётся слишком много специальностей, чтобы их можно было легко между собой сравнить, а если разбивать по слишком большому количеству критериев, то даже после того, как он выберет в каждой из групп лучший вариант, всё равно будет непонятно, как их сравнивать между собой. Поэтому Никита решил из всех критериев оставить только два, и все специальности разбить на две группы по первому выбранному критерию, после чего каждую из них снова разбить на две по второму. Понятно, что ему хочется, чтобы размер самой большой группы из полученных четырёх был как можно меньше. Осталось только понять, какие это должны быть критерии.

Input

В первой строке даны два целых числа n и m — количество специальностей на матмехе и количество выбранных Никитой критериев соответственно ($4 \leq n, m \leq 100$). В следующих n строках записано по m чисел 1 или 2. i -е число в j -й строке равно единице, если при разбиении по i -му критерию j -я специальность попадает в первую группу и двойке в противном случае.

Output

В первой строке выведите единственное целое число — размер максимальной группы при оптимальном разбиении. Во второй строке выведите два различных числа — номера критериев, которые должен выбрать Никита.

Example

standard input	standard output
6 4	2
2 2 2 1	1 3
2 2 1 2	
2 1 1 1	
1 1 1 1	
2 2 2 2	
1 2 2 1	

Problem C. Болеро

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Денис — большой фанат классической музыки, и потому не может и недели прожить, не сходяв хотя бы на один концерт в филармонию, где он всегда покупает билет на одно и то же, идеальное, с его точки зрения, место. В этом году Денис решил не полагаться на удачу и закупить билетов сразу на весь сезон и даже уже выбрал список концертов, на которые он хочет сходить. Но внезапно он столкнулся с проблемой того, как именно купить эти билеты. Дело в том, что в Свердловской филармонии есть понятие личного абонеента — возможность выбрать любые k или более концертов и купить на них единый абонемент, стоимость которого будет равна суммарной стоимости всех входящих в него билетов со скидкой p процентов. Причём вариантов таких абонементов несколько, с разными k и p . Кроме того, на некоторые концерты Денису, как студенту, полагается скидка. Но эта скидка действует только в том случае, если он покупает этот билет отдельно, вне абонеента. И теперь Дениса, как и любого другого бедного студента, интересует вопрос, как ему закупить билеты на все концерты, потратив как можно меньше денег. При этом он знает, что желающих сходить на концерт почти всегда больше, чем мест в зале, поэтому он не будет покупать более одного билета на один концерт даже если это позволит ему сэкономить.

Input

В первой строке даны два числа n и m — количество концертов, на которые хочет сходить Денис и количество разных вариантов абонементов в филармонии ($1 \leq n, m \leq 10^5$). В следующих n строках написаны по два числа s_i и d_i — цена билета на i -й концерт и размер скидки студентам в процентах ($100 \leq s_i \leq 50\,000$, $0 \leq d_i \leq 100$). В следующих m строках даны по два числа k_i и p_i — минимальное число концертов, которые можно объединить в i -й абонемент и скидка по нему в процентах ($2 \leq k_i \leq 10^5$, $1 \leq p_i \leq 100$). Все числа на входе целые.

Output

Выведите единственное число — минимальное количество денег, которые надо потратить Денису, чтобы купить билеты на все выбранные им концерты. Ответ должен быть дан с абсолютной или относительной погрешностью не более 10^{-6} .

Example

standard input	standard output
6 4	2680.00
500 0	
700 0	
300 0	
400 0	
500 50	
800 0	
5 10	
7 20	
9 30	
11 40	

Problem D. Все дороги ведут в сугроб

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Для Фёдора сегодняшний вечер особенный — впервые за много лет в Екатеринбурге даёт свой концерт легендарная немецкая рок-группа Decline. И, как назло, именно в этот вечер решил повалить снег, основательно замедлив движение по всем дорогам. А ехать до концертного зала Фёдору нужно через весь город. Немного спасает ситуацию снегоуборочная техника, после проезда которой движение по дорогам заметно оживляется. Правда, в то время, пока техника работает, на эту улицу лучше совсем не заезжать. И теперь Фёдору нужно с учётом всех этих факторов выбрать маршрут, при котором он попадёт на концерт как можно раньше, чтобы успеть занять места в первых рядах толпы фанатов.

Для каждой из дорог в городе Фёдор знает время, за которое эту дорогу можно проехать при нормальной погоде. А в снегопад скорость движения и, соответственно, время проезда зависят от того, сколько снега выпало на дорогу. Фёдор много раз ездил в снегопад и составил эмпирическое правило: если выехать на дорогу через t полных минут снегопада, то время проезда по этой дороге надо увеличить на t процентов относительно нормальной погоды, но не более чем в 100500 раз. Поскольку правило приближительное, то полученное время Фёдор округляет вверх до целого числа минут. Разумеется, непосредственно после проезда снегоуборочной техники по какой-либо дороге счёт минут для неё начинается заново — на ней практически не остаётся снега после уборки.

Фёдору удалось найти в интернете полный план проезда снегоуборочной техники, и он хочет выстроить свой маршрут таким образом, чтобы никогда не заезжать на дорогу, если в это время там уже работает техника и успевать съехать с дороги не позже того момента, когда она там появится. В крайнем случае всегда есть возможность подождать на перекрёстке.

Input

В первой строке даны два целых числа n и m — количество перекрёстков и дорог между ними соответственно ($1 \leq n, m \leq 10^5$). Дом Фёдора находится на первом перекрёстке, а концертный зал — на n -м. В i -й из следующих m строк идёт описание i -й дороги — три целых числа a_i , b_i и v_i — перекрёстки, которые она соединяет и время проезда по ней в минутах ($1 \leq a_i, b_i \leq n$, $1 \leq v_i \leq 10^6$). Все дороги в Екатеринбурге двусторонние, между любыми двумя перекрёстками есть не более одной дороги, никакая дорога не соединяет перекрёсток сам с собой. И, конечно же, по дорогам можно доехать от дома Фёдора до концертного зала.

В следующей строке дано единственное целое число k — количество дорог, по которым проедет снегоуборочная техника ($0 \leq k \leq 10^5$). В i -й из следующих k строк записано три целых числа p_i , l_i и r_i — номер дороги, время, когда на ней начнёт работать снегоуборочная техника и время, когда на ней закончит работать техника ($1 \leq p_i \leq m$, $0 \leq l_i < r_i \leq 10^9$). За вечер на одной и той же дороге снегоуборочная техника может работать более одного раза, но при этом не может быть такого, что в какой-то момент времени на одной дороге находится несколько бригад техники.

Output

Выведите единственное число — минимальное время, за которое Фёдор сможет доехать до концертного зала.

Example

standard input	standard output
4 3 1 2 10 2 3 10 3 4 10 1 2 10 15	38

Problem E. Тысяча чертят

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Саша любит ходить в кино. А также он любит сказочные миры и истории про драконов, спасающих принцев от ужасных принцесс, и поэтому в кино он чаще всего ходит смотреть мультфильмы. Вот только есть одна маленькая деталь, которая почти всегда портит впечатление от похода в кинотеатр — другие зрители. На сеансах с мультфильмами всегда куча детей, из-за которых даже своё место в зале сложно занять — ведь когда протискиваешься к своему сидению в конце ряда, каждый ребёнок, мимо которого ты проходишь, так и норовит ненароком пнуть. А когда таки усядешься, то любой проходящий мимо тебя к своему месту ребёнок может случайно облить тебя колой. В общем, приятного мало. Поэтому в последнее время Саша решил поступать несколько умнее. Теперь он, заходя в зал, не сразу идёт к своему месту. Вместо этого он смотрит какие места уже заняты, и, если, например, его место у прохода, а большинство мест ещё свободны, то он предпочтёт подождать в проходе, пока остальные зрители не займут свои места, чтобы им не приходилось протискиваться через него. И наоборот, если его место в самом конце ряда, то он предпочтёт сразу занять его, пока не пришли другие зрители, через которых потом придётся перелезть. Сейчас Саша пытается выработать оптимальную стратегию, которая минимизировала бы сумму количества зрителей, через которых ему придётся протискиваться к своему месту и количества зрителей, которые пройдут к своему месту через него. При построении таковой стратегии Саша пользуется хорошо известными ему по предыдущему опыту фактами:

- если в момент, когда он заходит в зал, какое-то место ещё свободно, то с вероятностью $p\%$ до начала сеанса его никто так и не займёт;
- зрители заходят в зал по одному и в совершенно случайном порядке;
- все зрители, в том числе и Саша, проходят к своему месту с левого края ряда;
- опоздавших зрителей в зал не пускают. Поэтому если в момент начала сеанса Саша всё ещё стоит в проходе, то он понимает, что пропускать больше никого не придётся и садится на своё место.

Зная всё это, а также места, занятые в зале в тот момент, когда туда заходит Саша, определите матожидание количества зрителей, мимо которых придётся пройти Саше (или им через него) до начала сеанса при оптимальной стратегии.

Input

В первой строке дано два целых числа n и p — количество мест в ряду и вероятность в процентах, что свободное место до начала сеанса никто не займёт ($1 \leq n \leq 1\,000$, $0 \leq p \leq 100$). Во второй строке дана строка из n символов, описывающая ряд слева направо в тот момент, когда Саша заходит в зал. i -й символ равен 0, если i -е слева место свободно, 1, если оно занято и 2, если это место Саши. Гарантируется, что в строке есть ровно один символ 2.

Output

Выведите одно число с абсолютной или относительной погрешностью не более 10^{-6} — ответ на задачу.

Example

standard input	standard output
6 0 011020	3.000

Problem F. Скрам

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Программист Иван любит работать. Настолько, что для него идеальный день — это когда он смог просидеть весь день до глубокой ночи в своём уголке в офисе, не отвлекаясь от написания кода на всякие посторонние дела. Жаль только, начальство не сильно его в этом поддерживает, всё время заставляя его участвовать во всяких беседах и презентациях.

С первого января нового года по всей организации было решено ввести новый порядок отчётов сотрудников перед руководителями. Каждый второй день каждый программист обязан отчитываться перед своим непосредственным начальником. Из оставшихся дней каждый третий — перед начальником своего начальника. Из оставшихся каждый четвёртый — перед начальником начальника своего начальника. И так далее. Организация очень крупная, потому можно считать, что для простых программистов в ней бесконечно много уровней начальства. И хотя вся эта бюрократия Ивана совсем не радует, деваться некуда — нужно учиться с ней жить. В частности, нужно спланировать отпуск так, чтобы на него выпало как можно больше отчётных дней и как можно меньше дней, в которые можно спокойно поработать. Иван хочет посчитать, сколько спокойных дней выпадает на период запланированного им отпуска, чтобы всё взвесить и, возможно, перенести отпуск на другое время.

Input

В первой строке даны два целых числа l и r — номера первого и последнего дней запланированного Иваном отпуска, считая со дня введения нового порядка ($1 \leq l \leq r \leq 10^9$).

Output

Выведите единственное число — количество спокойных дней в период запланированного отпуска.

Example

standard input	standard output
1 10	3

Note

В дни с номерами 2, 4, 6, 8 и 10 Иван будет отчитываться перед своим непосредственным начальником. В 5-й день — перед начальником своего начальника, в 9-й — перед начальником начальника своего начальника. Так что спокойными остаются только дни 1, 3 и 7.

Problem G. Всё включено

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Настя не любит зиму. Она любит яркое солнце и тёплое море. Поэтому она каждый год посреди зимы вместе со своими друзьями летит на пару недель в какую-нибудь из тропических стран поваляться на пляже и посмотреть местные достопримечательности. И каждый год она сталкивается с проблемой планирования поездки. Сама она для себя может легко запланировать, в какой день что она будет делать. А вот с её друзьями всё намного сложнее. Составить план поездки никто из них сам не в состоянии, поэтому каждый из них, договариваясь с Настей, сначала долго её расспрашивает о том, кто в какой из дней чем планирует заняться, после чего выбирает за основу один из планов, в нём меняет несколько дней под себя и на этом останавливается. А для Насти всё это выливается в одну большую головную боль — ведь нужно очень внимательно отслеживать, кто в итоге на какой день что для себя запланировал. И ладно бы если каждый под себя менял только день-два, но многие просто выдают просьбы вроде «Хочу выделить четыре дня поваляться на пляже. Причём не с начала отдыха, а с пятого дня. И не подряд, а через день. А ещё три дня — на походы в коктейль-бар. Начиная с седьмого дня раз в неделю». А Насте нужно из этого понять, что в пятый, девятый и в одиннадцатый день человек хочет поваляться на пляже, а в седьмой, четырнадцатый и двадцать первый — посидеть в баре. И за предыдущие годы это всё её настолько задолбало, что она решила в этом году несколько автоматизировать весь процесс.

Input

В первой строке два целых числа n и m — количество дней поездки и количество друзей Насти ($1 \leq n, m \leq 10^5$). Во второй строке через пробел даны n целых чисел a_i — планируемые виды деятельности в каждый из дней по плану Насти ($1 \leq a_i \leq 10^9$). Далее следуют m блоков, описывающих планы друзей Насти. В первой строке i -го блока идёт одно целое число q_i — количество разных вопросов, которые задаёт i -й друг перед окончательным выбором своего плана ($0 \leq q_i \leq 10^5$). В j -й из следующих q_i строк записано два целых числа f_{ij} и a_{ij} , которые означают вопрос, о том, какая активность запланирована на a_{ij} -й день f_{ij} -го плана ($0 \leq f_{ij} < i$, $1 \leq a_{ij} \leq n$). В следующей строке дано два целых числа p_i и c_i — номер плана, который взят за основу и количество изменений в нём ($0 \leq p_i < i$, $0 \leq c_i \leq n$). В следующих c_i строках записаны по четыре числа d_{ij} , k_{ij} , p_{ij} и t_{ij} — номер первого дня, в котором нужно поменять активность, общее количество дней, в которые нужно поменять активность, период между днями и вид активности, на которую нужно поменять ($1 \leq d_{ij}, k_{ij}, p_{ij} \leq n$, $d_{ij} + (k_{ij} - 1) \cdot p_{ij} \leq n$, $1 \leq t_{ij} \leq 10^9$). Сумма всех q_i на входе не превосходит 10^5 , сумма всех k_{ij} не превосходит $5 \cdot 10^6$, сумма всех c_i не превосходит 10^5 .

Output

Выдайте m строк, в i -й из них — q_i чисел через пробел — ответы на вопросы i -го друга.

Example

standard input	standard output
3 2	3
1 2 3	4 5 5
1	
0 3	
0 2	
1 2 2 4	
2 2 1 5	
3	
1 1	
1 2	
1 3	
0 0	

Problem H. Чем выше горы

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Катя предпочитает активный отдых, поэтому она регулярно со своими друзьями выезжает на один из ближайших горнолыжных курортов покататься на сноуборде. Сегодня она решила поехать на гору Шпильную, ведь там как раз открыли новую трассу. Но после того, как она прокатилась по этой трассе в первый раз, её посетило ощущение дежавю, как будто она до этого уже каталась по точно такому же склону. Ну, или, может быть, чуть более крутому или пологому. И почему-то её это настолько взволновало, что вместо того, чтобы просто продолжить кататься, она решила проверить, не было ли на каком-либо из склонов, по которым она каталась до этого, отрезка, который был бы похож на тот, по которому она съехала только что. Вернувшись в свою машину и достав ноутбук, Катя смогла найти в интернете подробную информацию о всех склонах, по которым она ездила — карту высот с шагом в один метр. Катя считает похожими два отрезка разных склонов одинаковой длины, если существуют такие числа a и b , что если мы обозначим высоты первого отрезка за x_0, x_1, \dots, x_n , а высоты второго — за y_0, y_1, \dots, y_n , то $x_i - y_i = a \cdot i + b$.

Input

В первой строке дано единственное целое число n — количество склонов, по которым Катя ездила до этого ($1 \leq n \leq 10^5$). Во второй строке через пробел даны целое число m и $m + 1$ целых чисел $x_0 \dots x_m$, где m — длина склона, по которому Катя проехала только что, а x_i — высота над уровнем моря точки, находящейся в i метрах от начала склона ($1 \leq m \leq 10^5$, $-10^9 \leq x_i \leq 10^9$). В следующих строках в точно таком же формате заданы все склоны, по которым Катя каталась до этого. Гарантируется, что сумма их длин не превосходит 10^5 .

Output

Если существуют два таких числа i и j , что склон, по которому только что проехала Катя, похож на отрезок, начинающийся с j -го метра от начала i -го склона из тех, по которым она ездила ранее, то выведите числа i и j через пробел. Если есть несколько таких пар i и j , то выведите ту, при которой значение модуля параметра a из критерия похожести минимально. Если таких всё ещё несколько, выведите любую. Если таких чисел не существует, то выведите -1 .

Examples

standard input	standard output
2 2 3 2 1 5 21 15 10 6 3 2 4 10 7 5 3 2	2 1
3 2 0 0 0 1 0 0 2 1 2 4 1 5 17	-1

Problem I. Чем ниже горы

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротаёт долгие зимние вечера по-своему. Макс любит гонять по ночному городу на своей тонированной Приоре с заниженной подвеской. И ничто его не может остановить — ни снегопад, ни гололёд. Ну разве что фонарный столб или автобусная остановка. Или, как сегодня, бампер чёрного Мерседеса с четырьмя крепкими парнями внутри, которые не сильно обрадовались влетевшему в их машину сзади ведру с болтами. И теперь перед Максом стоит сложный выбор. Он может договориться на месте и просто заплатить за нанесённый ущерб; благо, Мерседес — машина крепкая, и все его повреждения — лишь несколько царапин. А может вызвать наряд ГИБДД и оформить ДТП как положено. В таком случае за всё заплатит страховая компания, но стоимость страховки для Макса в ближайшие несколько лет будет явно выше. И поскольку считает Макс достаточно плохо, а ребята из Мерса уже проявляют явное недовольство, Макс решил позвонить вам, чтобы вы решили, который из двух вариантов для него выгоднее. И лучше вам поторопиться с решением.

В текущем году Макс платит за страховку c рублей. Если он не будет оформлять ДТП, то за следующий год он заплатит $c + d$ рублей, затем — $c + 2d$ рублей, и так далее, каждый следующий год стоимость страховки будет расти на d рублей. Если же оформить ДТП, то k лет, начиная со следующего, Максусу придётся платить на p процентов больше по сравнению со стоимостью обычной страховки.

Input

В первой строке дано единственное целое число b — сумма в рублях, на которую Макс может договориться с водителем Мерседеса ($1 \leq b \leq 10^5$). Во второй строке даны два целых числа c и d — текущая стоимость страховки в рублях и на сколько рублей она увеличивается каждый год, если не попадать в ДТП ($1 \leq c, d \leq 10^4$). В третьей строке даны два целых числа k и p — сколько лет Максусу придётся покупать более дорогую страховку и на сколько процентов её стоимость выше стоимости обычной страховки ($1 \leq k, p \leq 100$).

Output

Если Максусу выгоднее договориться на месте, то в первой строке выведите слово “Cash”, иначе выведите слово “Insurance”. Во второй строке выведите единственное число — то, на сколько рублей выбранный вариант оказывается выгоднее второго. Ответ должен быть дан с абсолютной или относительной погрешностью не более 10^{-6} .

Examples

standard input	standard output
10000 5000 1000 3 50	Cash 500.00
15000 2500 1000 4 25	Insurance 10000.00

Problem J. Space invader

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Зима в Екатеринбурге — самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Например, Алексей создаёт по вечерам компьютерные игры. Нынешняя его игра имеет весьма простой игровой процесс — пользователь управляет космическим кораблём, летящим в пространстве. Для тестирования механики движения корабля Алексей хочет решить такую задачу: может ли его корабль, двигаясь по прямой, пролететь через точки A и B, после чего совершить поворот на 90° и, продолжая двигаться по прямой, пролететь ещё через точки C и D. Точки A, B, C, D должны быть посещены кораблём именно в таком порядке, точка поворота может совпадать с точкой B или точкой C. Корабль следует считать материальной точкой.

Input

В четырёх строках дано по три целых числа x_i, y_i, z_i — координаты точек A, B, C, D, соответственно ($-10^6 \leq x_i, y_i, z_i \leq 10^6$). Все точки попарно различны.

Output

Если корабль может пролететь через данные точки, выведите “Valid”, иначе выведите “Invalid”.

Example

standard input	standard output
-2 0 0 -1 0 0 0 1 0 0 2 0	Valid