Задача A. Who Calls the Crystal Maiden? (Юниорская лига)

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася тренируется играть в Dota2. Сейчас он играет за героя Crystal Maiden — юную жрицу холода. Самым сильным заклинанием героя является Freezing Field. Во время действия заклинания вокруг Crystal Maiden происходят ледяные взрывы. Каждый взрыв наносит урон всем врагам на расстоянии не более r от центра взрыва. Центр каждого взрыва находится на расстоянии не более R от Crystal Maiden, причем местоположение центра взрыва равновероятно распределено в соответствующем круге. Всего происходит N взрывов. Взрывы независимы между собой, то есть положение одного взрыва никак не сказывается на положении следующих взрывов. Вася встретил вражеского героя и знает, что для убийства этого героя необходимо, чтобы его затронули хотя бы K взрывов заклинания Freezing Field. Зная положение вражеского героя, определите вероятность того, что он будет убит Васей.

Формат входного файла

Первая строка содержит два целых числа: $1 \le r, R \le 1000$ — радиус взрыва и радиус действия заклинания соответственно. Во второй строке находятся два целых числа $1 \le N, K \le 1000$ — общее число взрывов при применении заклинания и число взрывов, необходимое для убийства вражеского героя. В последней строке находятся два целых числа $-1000 \le X, Y \le 1000$ — координаты вражеского героя относительно Crystal Maiden.

Формат выходного файла

Необходимо вывести единственное вещественное число с абсолютной погрешностью не более 10^{-6} — вероятность того, что вражеский герой будет убит в результате применения заклинания.

Пример

stdin	stdout
2 10 1 1 2 3	0.0400000000
2 10 2 1 2 3	0.0784000000

Задача B. World of Dota: Cross (Юниорская лига)

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася играет в Dota 2, но ему надоело воевать каждый раз на одной и той же карте. Поэтому Вася решил сделать новую карту для этой игры. Для простоты будем полагать, что карта представляет собой прямоугольник 9×9, каждая клетка которого может быть либо свободна, либо занята деревом. Вася хочет, чтобы вражеская команда (состоящая из пяти героев) не могла группироваться близко к одной точке. Поэтому на карте не должно быть пяти свободных клеток, образующих крест с шириной и высотой 3:

```
.*.
***
.*.
```

С другой стороны, Вася хочет оставить как можно больше пространства для маневров, поэтому деревьев должно быть как можно меньше. Помогите Васе найти идеальную карту.

Формат входного файла

Эта задача не содержит входных данных.

Формат выходного файла

Необходимо вывести 9 строк, по 9 символов в каждой строке — искомую карту для Васи. Свободная клетка должна обозначаться символом '.', занятая деревом — '#'. Ответ считается верным, если карта является допустимой (не содержит крестов, состоящих из пяти свободных клеток) и содержит наименьшее число деревьев среди всех возможных допустимых карт.

Пояснение

Пример оптимальной допустимой карты 4×4:

```
....
.#..
..#.
....
```

Задача C. Warlock (Юниорская лига)

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася тренируется играть в Dota2. Сейчас он играет за героя Warlock. Одной из способностей героя является Fatal Bounds — герой «связывает» несколько вражеских юнитов вместе, заставляя их получать дополнительный урон. Заклинание работает следующим образом: если один из связанных юнитов получает урон (основной урон), то все остальные юниты получают P процентов от полученного юнитом урона (дополнительный урон). Действие связки не распространяется на дополнительный урон, полученный в результате действия заклинания (дополнительный урон не вызывает получение дополнительного урона остальными юнитами). Юнит, получивший основной урон, не получает дополнительный урон. Если юнит погибает при получении основного урона (количество очков жизни становится неположительным), то дополнительный урон будет рассчитываться от количества единиц жизни юнита перед получением основного урона, а не от изначального значения получаемого урона.

Вася хочет убить вражеского героя, поэтому связал его с N крипами при помощи Fatal Bounds. Далее будем считать, что вражеский герой не получает основной урон, в то время как все крипы будут воевать с Васей, пока не будут убиты. При этом часть полученного ими урона будет переходить Васе как дополнительный урон. Разумеется, сами крипы тоже будут страдать от получения дополнительного урона от Fatal Bounds. Вася может атаковать крипов произвольное число раз, при этом каждый раз он выбирает одного живого крипа, которому будет наноситься урон, и величину урона (положительное число). Вася атакует крипов, пока они не будут убиты. Зная изначальное количество очков здоровья у крипов, определите, какой наибольший урон может при этом получить вражеский герой.

Формат входного файла

В первой строке входного файла содержатся два целых числа: N — количество крипов, связанных с вражеским героем, $1 \le N \le 100000$ и P — процент дополнительного урона относительного основного, $0 \le P \le 100$. Во второй строке содержатся N натуральных чисел, не превосходящих 1000 — количество единиц здоровья у крипов, связанных Fatal Bounds.

Формат выходного файла

Необходимо вывести одно вещественное число с абсолютной или относительной погрешностью не более 10^{-6} — наибольший урон, который может получить вражеский герой.

Пример

stdin	stdout
2 20 100 10	21.6000000000

Задача D. Wild Card: Subway

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася долго учился играть в Dota2, но его команда так и не смогла пробиться на финал Starladder 2014. Вася был жутко расстроен и подумывал о завершении карьеры киберспрортсмена, как вдруг ему позвонил однокомандник и сообщил радостное известие: организаторы финала предоставили команде Васи Wild Card (дополнительное место) для участия в турнире. Но возникла другая проблема — до начала финала осталось не так уж много времени, Васе срочно нужно попасть на вокзал и прибыть к месту проведения. Самый быстрый способ добраться до вокзала — воспользоваться местным метрополитеном. Вася забежал в метро и начал искать возможность приобрести билет для проезда. К счастью, на станциях метро недавно были установлены автоматы по продаже билетов. Автомат действует следующим образом. Сначала автомат принимает некоторое количество монет. Если этого количества недостаточно, то автомат ничего не делает (да, получить монеты назад невозможно). Если монет достаточно для покупки одного билета, то автомат рассчитывает размер сдачи. Если автомат может выдать требуемую сдачу (пользуясь уже находящимися в нем монетами), то он выдает билет и сдачу. В противном случае, он переходит в состояние ошибки и ничего не выдает, даже билет.

Однако вернемся к Васе. Вася собрался бросить несколько монет в автомат, но задумался — не случится ли так, что он останется без билета? Зная наборы монет, находящиеся в автомате и у Васи, определите, может ли Вася бросить в автомат часть своих монет (возможно — все монеты) так, что их будет достаточно для покупки билета, но автомат не сможет выдать сдачу.

При выдаче сдачи автомат также может использовать монеты, которые в него бросил Вася. Гарантируется, что у Васи хватает денег на покупку билета.

Формат входного файла

В первой строке даны три натуральных числа: N — количество монет у Васи, K — количество монет в автомате к моменту прихода Васи и M — стоимость билета метро ($1 \le N, K, M \le 5\,000$).

Во второй строке даны N натуральных чисел — номиналы монет Васи.

В третьей строке даны K натуральных чисел — номиналы монет, находящихся в автомате.

Монеты по номиналу не превосходят 10 000. Наборы монет заданы в неубывающем порядке по номиналу.

Формат выходного файла

Если Вася сможет получить билет, бросив в автомат любую сумму, достаточную для покупки жетона, то выведите слово Go. Если же Вася может перевести автомат в состояние ошибки, то выведите слово Wait.

Пример

stdin	stdout
3 2 4 1 2 3 1 2	Go
3 2 4 1 2 3 2 4	Wait

Задача E. Wild Card: Bus

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Не сумев приобрести билет в метро (см. задачу «Wild Card: Subway»), Вася решил воспользоваться автобусом для поездки на вокзал. Город, в котором живет Вася, представляет собой множество остановок и двусторонних дорог, связывающих остановки. Автобус может перемещаться по дорогам между остановками. Для каждой дороги известно, сколько времени требуется автобусу для преодоления ее. Кроме того, автобус может находиться на остановке произвольное неотрицательно количество времени.

Вася уже давно играет в Dota2, благодаря этому он стал известным человеком в своем городе. Один из фанатов Васи, Петя, утверждает, что сумел получить автограф Васи на одной из остановок города. Точное время получения автографа Петя не помнит, но утверждает, что это произошло в какой-то момент между времени t_b и t_e. Не все жители города поверили словам Пети. Для того, чтобы опровергнуть слова Пети, некоторые даже стали изучать показания камер наблюдения, установленных на некоторых остановках. Зная времена, требуемые для проезда между остановками, и показания камер наблюдения определите, мог ли Вася дать автограф Пете на указанной остановке в предполагаемый интервал времени.

Формат входного файла

В первой строке находятся два числа N, M — количество остановок в городе и количество дорог соответственно $\left(1 \le N \le 1\,000, 0 \le M \le \frac{N(N-1)}{2}\right)$.

В следующих M строках описывается карта города. В каждой строке даны числа a_i, b_i, l_i $(1 \le a_i < b_i \le N, 1 \le l_i \le 10\,000)$ — номера остановок, соединенных дорогой, и время, нужное автобусу для перемещения по этой дороге. Каждая пара остановок соединена не более чем одной дорогой. По любой дороге можно двигаться в обе стороны.

В следующей строке записаны три целых числа C, t_b и t_e $(1 \le C \le N, 0 \le t_b \le t_e, \le 15\,000\,000)$ — номер остановки, указанной Петей, самое раннее и самое позднее время (включительно), в которое Вася мог оставить автограф.

В следующей строке находится одно число K $(1 \le K \le 50\,000)$ — количество показаний с камер наблюдения.

В следующих K строках записаны пары целых чисел p_j, t_j $(1 \le p_j \le N, 0 \le t_j \le 15\,000\,000)$ — номер остановки и время, когда камера зафиксировала Васю, едущего в автобусе. Все временные отметки t_j различны. Водитель автобуса не так торопится на вокзал, как Вася, поэтому его путь не является оптимальным, более того, автобус может проезжать по одним и тем же дорогам несколько раз в процессе одного рейса.

Формат выходного файла

Если Вася мог дать автограф Пете в указанный интервал времени на указанной остановке, то выведите `YES`, в противном случае — выведите `NO`.

Пример

stdin	stdout
4 6 1 2 10 2 3 10 3 4 10 1 4 10 1 3 15 2 4 15 2 12 25 2 1 10 3 30	YES
4 6 1 2 10 2 3 10 3 4 10 1 4 10 1 3 15 2 4 15 2 12 15 2 1 10 3 25	NO

Задача F. Windrunner at Your Service

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася тренируется играть в Dota2. Сейчас он играет за героя Windrunner. Одна из способностей героя — Shackleshot — позволяет привязать противника к дереву. Для осуществления привязки необходимо выполнение следующих условий:

- расстояние от героя до врага не должно превышать $D1$
- расстояние от врага до дерева не должно превышать $D2$
- угол между векторами «герой-враг» и «враг-дерево» не должен превышать A градусов по абсолютному значению.

Вася знает положение своего героя (X_w, Y_w), положение врага (X_e, Y_e), а так же положение всех N деревьев на карте (X_i, Y_i). Считается, что враг неподвижен. Помогите Васе определить, какое наименьшее расстояние придется преодолеть его герою для того, чтобы привязать врага к одному из деревьев.

Формат входного файла

Первая строка входного файла содержит три целых числа — $D1, D2$ и A ($1 \le D1, D2 \le 1000$, $1 \le A \le 90$). Во второй строке содержатся четыре целых числа — X_w, Y_w, X_e, Y_e. В третьей строке находится единственное натуральное число N ($1 \le N \le 1000$) — количество деревьев расположенных на карте. В следующих N строках находятся координаты деревьев — X_i, Y_i. Все координаты (героя, врага и деревьев) являются целыми числами и не превосходят 1000 по абсолютному значению. Ни одно дерево не находится в одной точке с вражеским героем.

Формат выходного файла

Если Вася может привязать врага к какому-либо дереву, то необходимо вывести минимальный путь, который придется проделать его герою, с абсолютной или относительной погрешностью не более 10^{-6}. В противном случае необходимо вывести -1.

Пример

stdin	stdout
1 2 30 0 0 1 1 1 2 2	0.4142135624
1 2 30 0 0 1 1 1 3 3	-1
2 2 30 0 0 1 1 1 2 2	0.0000000000

Задача G. What is the Answer?

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася хорошо научился играть в Dota2. Теперь Вася решил участвовать в Challenge 24 — популярном соревновании по программированию. На отборочном раунде ему встретилась задача, очки за которую начисляются в зависимости от того, насколько эффективно решение участника по отношению к эффективности лучшего отправленного кем-либо решения. Эффективность решения оценивается одним натуральным числом A — абсолютным показателем решения. Чем меньше это число, тем лучше решение. Баллы за задачу (относительный показатель решения) начисляются по следующей формуле:

$$R = [100 * (1 - \sqrt{(1 - BA/A)})]$$

где BA — абсолютный показатель лучшего отправленного решения, а A — абсолютный показатель решения участника, а $[X]$ — целая часть числа X. В частности, из формулы следует, что если участник отправил наилучшее решение, то он получит 100 баллов.

По правилам соревнования, абсолютный показатель наилучшего решения не сообщается до конца раунда. Однако, каждому участнику сообщается абсолютный и относительный показатель его решения. Вася отправил решение а абсолютным показателем A и получил R баллов (относительный показатель). Помогите Васе узнать наименьшее и наибольшее возможные значения абсолютного показателя наилучшего решения.

Формат входного файла

В единственной строке входного файла содержатся два натуральных числа — A и R, $1 \le A \le 10^9$, $1 \le R \le 100$.

Формат выходного файла

Если ни при каком значении абсолютного показателя наилучшего решения Вася не мог получить такие баллы, то выведите -1. В противном случае необходимо вывести 2 числа — наименьший и наибольший возможные абсолютные показатели наилучшего решения.

Пример

stdin	stdout
100 99	-1
100 50	75 75
100 10	19 20

Задача H. We Admit No Defeat

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася тренируется играть в Dota2. Сейчас он играет за героя Mirana. Впрочем, в данной задаче это неважно — герои не будут пользоваться своими специальными способностями. Васе предстоит сразиться один на один с вражеским героем. Разумеется, Вася не приемлет поражение! Битва будет честной, поэтому герои будут просто наносить удары друг другу. Можно считать, что скорость атаки героев одинакова и они наносят удары одновременно. Герои будут наносить удары, пока один из них (или оба) не погибнет. Несмотря на огромное желание победить, герои подчиняются игровой механике. Для поединка важны следующие параметры героев:

- Количество очков здоровья
- Базовый урон
- Броня

При нанесении удара урон считается по следующей формуле $D = BD/(1 + A * 0.06)$, где D — урон, нанесенный герою, BD — базовый урон атакующего героя, A — броня героя, получившего удар. Итоговое значение урона не округляется и вычитается из количества очков здоровья героя, получившего удар. Если количество очков здоровья в результате удара станет неположительным, то считается, что герой убит. Поскольку два героя атакуют одновременно, то может случиться ситуация, при которой оба героя погибают. Базовый урон каждого героя строго положителен, поэтому битва будет конечной.

Кроме того, герои могут покупать предметы, которые увеличивают одну (или несколько) характеристик. Каждый герой может купить максимум 6 предметов, один и тот же предмет можно покупать несколько раз. После того, как предметы куплены, начинается битва. Помогите Васе определить, сможет ли он выжить в битве?

Формат входного файла

Первая строка входного файла содержит три числа — начальное количество очков здоровья, базовый урон и броня героя Васи. Во второй строке приводится описание вражеского героя в том же формате. В следующей строке находится единственное число N ($1 \le N \le 20$) — количество возможных предметов для покупки. Далее идет описание предметов. Первая строка описания содержит название предмета (непустая строка, состоящая не более, чем из 50 букв латинского алфавита) и число K ($1 \le K \le 3$) — количество бонусов от предмета. В следующих K строках находится описание бонусов. Каждое описание бонуса имеет вид:

<BonusName> +X

где <BonusName> — тип изменяемой характеристики:

- 'HP' — для количества очков здоровья
- 'Damage' — для базового урона
- 'Armor' — для брони

X — численное значение бонуса к характеристике. Все бонусы от одного предмета имеют разные значения <BonusName>. Все характеристики (как начальные характеристики героев, так и бонусы предметов) — целые неотрицательные числа, не превосходящие 1000. Бонусы от предметов не могут быть нулевыми. Начальное здоровья героя строго положительно. Все названия предметов различны. Смотрите пример для уточнения формата входных данных.

Формат выходного файла

Если для любого множества купленных вражеским героем предметов существует множество предметов, при покупке которого герой Васи останется жив, то нужно вывести 1. Если существует множество предметов, при покупке которого вражеским героем Вася умрет независимо от того, какие купит предметы своему герою, то следует вывести -1.

Пример

stdin	stdout
600 40 1 450 40 8 1 IronBranch 2 Damage +1 HP +19	-1
600 40 1 450 40 8 2 IronBranch 2 Damage +1 HP +19 Chainmail 1 Armor +5	1

Задача I. Wex-Wex-Wex

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася играет в Dota 2. Одним из самых сложных и интересных героев в игре является Invoker. Как и любой другой настоящий герой, Invoker должен убивать! В ряд перед героем выстроились N крипов (неуправляемых юнитов), у каждого из которых ровно H единиц здоровья. Крипы расположены в точках с координатами от 1 до N по одному крипу в каждой целочисленной точке. Invoker может убивать крипов при помощи своих заклинаний. На самом деле, он может использовать 10 различных заклинаний, но в данной задаче он будет использовать только 3 из них.

- Chaos Meteor — наносит одинаковый фиксированный урон всем крипам на некотором отрезке.
- Sun Strike — наносит фиксированный урон, который делится поровну на всех *живых* крипов на некотором отрезке. Если при делении урона получается нецелое число, то оно округляется вниз до целого.
- Tornado — поднимает крипов в воздух на некоторое время, делая их неуязвимыми к заклинаниям, после чего опускает на землю и наносит одинаковый фиксированный урон каждому из поднятых крипов. Как и предыдущие заклинания, действует на некотором отрезке.

Крип считается мертвым, если количество единиц его здоровья после очередного заклинания стало не больше 0. Поднятые в воздух заклинанием Tornado крипы не считаются за живых при расчете урона от заклинания Sun Strike. Если заклинание Tornado заканчивает свое действие в момент времени `T` и в этот же момент времени используется другое заклинание, то сначала наносится урон от Tornado, а потом начинается действие другого заклинания.

По заданному набору использованных заклинаний определите, какое суммарное количество урона мог нанести Invoker по крипам.

Формат входного файла

В первой строке даны три натуральных числа N — количество крипов и H — начальное количество единиц здоровья каждого из крипов и M — количество примененных заклинаний. $1 \le N, M \le 100\,000$, $1 \le H \le 1\,000\,000\,000$.

В следующих M строках описаны примененные заклинания, по одному в строке в порядке их использования героем. Описание заклинания имеет вид:

`<Type> <Time> <L> <R> <D> [<T>]`, где

`<Type>` — тип примененного заклинания (1 — Chaos Meteor, 2 — Sun Strike, 3 — Tornado);

`<Time>` — время использования заклинания;

`<L>` — левая граница действия заклинания (включительно);

`<R>` — правая граница действия заклинания (включительно);

`<D>` — урон заклинания (при использовании Sun Strike делится между всеми живыми крипами);

`<T>` — время действия (только при заклинании Tornado);

$1 \le \mathrm{Time}, T \le 1\,000\,000\,000$; $1 \le L \le R \le N$; $1 \le D \le 1\,000\,000\,000$.

Заклинания затрагивают всех крипов на отрезке от `<L>` до `<R>` включительно. Все переменные времени `<Time>` различны. Гарантируется, что в любой момент времени «активно» не более одного Tornado.

Формат выходного файла

Вывести единственное число — суммарный урон, нанесенный героем.

Пример

stdin	stdout
5 10 7 1 1 2 5 4 1 2 2 2 8 2 3 1 3 9 3 4 1 4 2 2 2 5 3 5 4 2 6 3 5 4 1 7 5 5 6	44

Задача J. Warlock-2 (Высшая лига)

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася тренируется играть в Dota2. Сейчас он играет за героя Warlock. Одной из способностей героя является Fatal Bounds — герой «связывает» несколько вражеских юнитов вместе, заставляя их получать дополнительный урон. Заклинание работает следующим образом: если один из связанных юнитов получает урон (основной урон), то все остальные юниты получают P процентов от полученного юнитом урона (дополнительный урон). Действие связки не распространяется на дополнительный урон, полученный в результате действия заклинания (дополнительный урон не вызывает получение дополнительного урона остальными юнитами). Юнит, получивший основной урон, не получает дополнительный урон. Если юнит погибает при получении основного урона (количество очков жизни становится неположительным), то дополнительный урон будет рассчитываться от количества единиц жизни юнита перед получением основного урона, а не от изначального значения получаемого урона.

Вася хочет убить вражеского героя, поэтому связал его с N крипами при помощи Fatal Bounds. Далее будем считать, что вражеский герой не получает основной урон, в то время как все крипы будут воевать с Васей, пока не будут убиты. При этом часть полученного ими урона будет переходить Васе как дополнительный урон. Разумеется, сами крипы тоже будут страдать от получения дополнительного урона от Fatal Bounds. Вася может атаковать крипов произвольное число раз, при этом каждый раз он выбирает одного живого крипа, которому будет наноситься урон, и величину урона (положительное число). Вася атакует крипов, пока они не будут убиты. Зная изначальное количество очков здоровья у крипов, определите, какой наименьший и какой наибольший урон может при этом получить вражеский герой.

Формат входного файла

В первой строке входного файла содержатся два целых числа: N — количество крипов, связанных с вражеским героем, $1 \le N \le 100000$ и P — процент дополнительного урона относительного основного, $0 \le P \le 100$. Во второй строке содержатся N натуральных чисел, не превосходящих 1000 — количество единиц здоровья у крипов, связанных Fatal Bounds.

Формат выходного файла

Необходимо вывести два вещественных числа с абсолютной или относительной погрешностью не более 10^{-6} — наименьший и наибольший урон, который может получить вражеский герой.

Пример

stdin	stdout
2 20 100 10	20.0000000000 21.6000000000

Задача K. World of Dota: Cross 2 (Высшая лига)

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася играет в Dota 2, но ему надоело воевать каждый раз на одной и той же карте. Поэтому Вася решил сделать новую карту для этой игры. Для простоты будем полагать, что карта представляет собой прямоугольник $N \times N$, каждая клетка которого может быть либо свободна, либо занята (деревом или водой). Вася хочет, чтобы вражеская команда (состоящая из пяти героев) не могла группироваться близко к одной точке. Поэтому на карте не должно быть пяти свободных клеток, образующих крест с шириной и высотой 3:

```
.*.
***
.*.
```

Для того, чтобы предотвратить появление таких крестов, Вася может сажать деревья в свободные клетки. Кроме того, вдоль диагонали карты течет вода. Это значит, что каждая клетка главной диагонали исходного прямоугольника $N \times N$ занята водой. Сажать деревья на воду нельзя, но и вражеские герои на нее не встают.

Вася хочет оставить как можно больше пространства для маневров, поэтому деревьев на карте должно быть как можно меньше. Вася уже расставил несколько деревьев на карте. Помогите Васе расставить еще как можно меньше деревьев так, чтобы пять вражеских героев не могли образовать крест.

Формат входного файла

В первой строке находится единственное натуральное число — N — размер карты ($1 \le N \le 17$). Далее следуют N строк по N символов в каждой — описание карты. Символ '.' соответствует свободной клетке, символ '#' — занятой (деревом или водой). Гарантируется, что все клетки главной диагонали заняты.

Формат выходного файла

Необходимо вывести описание новой карты — N строк по N символов в каждой. Если клетку следует оставить свободной, то надо выводить символ '.', если клетка изначально была занята — символ '#', если в изначально свободную клетку нужно добавить дерево — символ '+'. Если оптимальных решений несколько, можно вывести любое из них.

Пример

stdin	stdout
6 #..... .#.... ..#... ...#.. ....#. .....#	#..... .#..+. ..#... ...#.. .+..#. .....#
6 #..... .#.... ..#... ...#.. .#..#. .....#	#..... .#..+. ..#... ...#.. .#..#. .....#

Задача L. Wild Card: Plane (Высшая лига)

Вход:	stdin
Выход:	stdout
Ограничение по времени:	2 с
Ограничение по памяти:	256 Мб

Вася успешно добрался до вокзала, сел на аэроэкспресс и прибыл в аэропорт. Теперь Вася готов лететь на финал Starladder 2014. Однако, еще одно неприятное событие может помешать Васе прибыть на финал вовремя. Давным давно потухший вулкан вдруг проснулся и начал выбрасывать в воздух потоки золы и дыма. Несколько самолетов уже упали в непосредственной близости от вулкана, потеряв управление из-за вулканических пород, оказавшихся в воздухе. Для безопасного полета, самолет должен всегда находиться на расстоянии не менее r (вдоль поверхности Земли) от вулкана. Таким образом, часть воздушного пространства в виде круга с центром в местоположении вулкана закрыта для перелетов. Считая, что Земля имеет форму шара радиуса R, а так же зная координаты вулкана, начальной и конечной точек полета, определите, какое наименьшее расстояние придется преодолеть самолету, чтобы безопасно осуществить свой рейс. Высотой полета самолета относительно Земли нужно пренебречь — считается, что самолет летит на малой высоте вдоль поверхности.

Формат входного файла

Первая строка содержит два целых числа — R и r — радиус Земли и расстояние от вулкана до границы закрытой зоны. $1 \le r, R \le 10^9$, гарантируется, что площадь поверхности закрытой зоны не превышает половины площади поверхности Земли. Во второй строке даны координаты вулкана — Lat_v и $Long_v$ — широта и долгота в градусах ($-90 \le Lat_v \le 90$, $-180 \le Long_v \le 180$). В третьей и четвертой строках даны координаты начальной и конечной точек полета в том же формате. Координаты всех точек целые. Гарантируется, что точки взлета и посадки различны и не лежат внутри или на границе закрытой зоны.

Формат выходного файла

Необходимо вывести единственное число — наименьшую длину пути безопасного полета вдоль поверхности Земли.

Пример

stdin	stdout
10 1 0 0 0 90 90 0	15.7079632679
10 1 0 0 0 45 0 -45	15.8083481818