

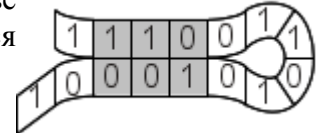
Для всех задач:

Имя входного файла: *input.txt*
Имя выходного файла: *output.txt*
Ограничение по памяти: *256 Мб*

Задача 1. Склеивание подстрок

Ограничение по времени на 1 тест: *2 сек.*
Java *3 сек.*

Дана строка S , состоящая из символов 0 и 1. Назовем две непересекающиеся подстроки S склеиваемыми, если одна может получиться из другой разворотом и заменой '0' на '1' и '1' на '0' (см. иллюстрацию).



Необходимо для заданной строки определить максимальную длину ее склеиваемых участков.

Входные данные

В единственной строке входного файла записана непустая строка S , состоящая из нулей и единиц. Длина строки не превосходит 100000.

Выходные данные

В единственную строку выходного файла необходимо вывести максимальную длину склеиваемых отрезков заданной строки. Если такой пары отрезков не существует, то вывести 0.

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|------------------|-------------------|
| 111001101010001 | 4 |

Задача 2. Джеймс Бонд

Ограничение по времени на 1 тест:

1 сек.

50-летию выхода первого фильма об агенте 007 посвящается...

В кинофильме «Огненный шар» британского агента Джеймса Бонда подбирают из лодки в океане на борт самолета следующим образом: Бонд надувает гелием воздушный шар и выпускает его в воздух на тросе. Второй конец троса он прицепляет себе к поясу. На носу самолета размещена специальная рогулька. Самолет зацепляет трос рогулькой и выдергивает Бонда из лодки.

Процесс затаскивания Бонда на борт самолета в фильме подробно не показан.

Вам необходимо написать программу, при помощи которой штурман самолета мог бы подобрать скорость и высоту, при которых эта операция не представляла бы опасности для жизни агента 007.

Входные данные

В единственной строке входного файла записаны два целых числа V и h — скорость самолета в километрах в час и высота полета самолета в метрах над уровнем моря ($100 \leq V \leq 1000$, $20 \leq h \leq 10000$).

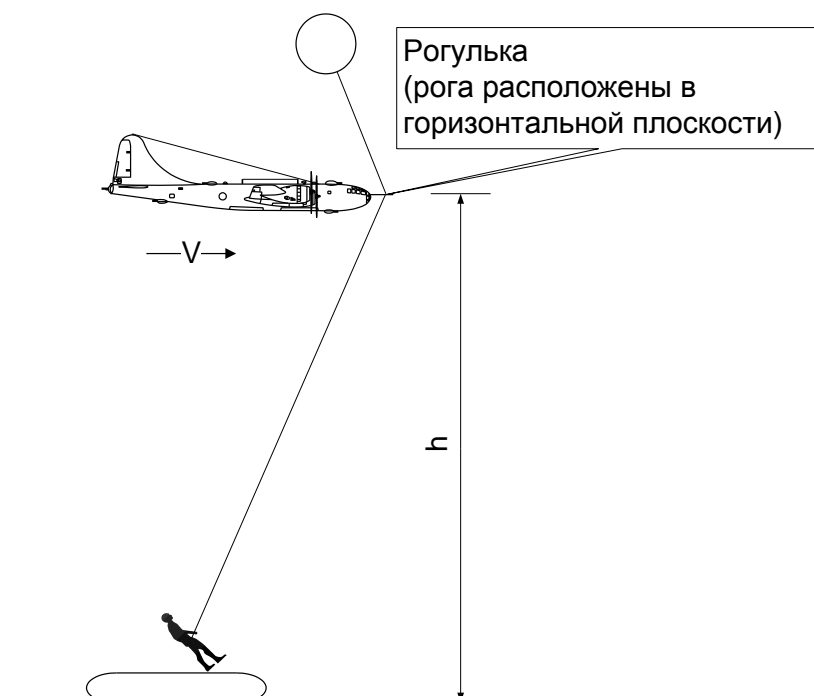
Выходные данные

Вам необходимо вывести в выходной файл целое число — ускорение Джеймса Бонда в момент сразу после соприкосновения рогульки с тросом. Значение ускорения измеряется в м/сек^2 и округляется до ближайшего целого числа.

При расчетах следует предполагать, что трос невесомый, нерастяжимый, не проскальзывает по рогульке, не испытывает сопротивления воздуха и в начальный момент висит строго вертикально. Также следует предполагать, что суммарная масса Бонда и воздушного шара пренебрежимо мала по сравнению с массой самолета, поэтому скорость самолета при соприкосновении с тросом не меняется.

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|------------------|-------------------|
| 500 100 | 193 |



Задача 3. Построение булевой функции

Ограничение по времени на 1 тест:

5 сек.

Java

10 сек.

Даны булева функция $F(a,b,c,d) = (a \wedge b) \vee (c \wedge d)$ и таблица истинности некоторой функции G — ее значения на всех входах в лексикографическом порядке.

Требуется написать программу, которая для данной таблицы истинности G выводит вариант построения функции G как композицию вызовов функции F .

Входные данные

В первой строке входного файла записано число T — количество тестов во входном файле ($1 \leq T \leq 10000$). Каждый тест задан двумя строками, в первой из которых задано единственное число N — количество аргументов функции G , а во второй строке — 2^N чисел 0 или 1, записанных через пробел, представляющих таблицу истинности функции G ($1 \leq N \leq 5$).

Выходные данные

Для каждой функции, описанной во входном файле, вывести без пробелов ее представление через F , минимальное по длине. Если такого представления не существует, то выведите число 0. Если минимальных представлений несколько — выведите любое.

Примеры

| <i>input.txt</i> | <i>output.txt</i> |
|------------------|-----------------------------|
| 2 | $F(a, F(a, b, a, c), b, c)$ |
| 3 | 0 |
| 0 0 0 1 0 1 1 1 | |
| 1 | |
| 1 0 | |

Комментарии

Для первого примера таблица истинности выглядит следующим образом:

| a | b | c | $G(a, b, c)$ | $F(a, b, a, c)$ | $F(a, F(a, b, a, c), b, c)$ |
|---|---|---|--------------|-----------------|-----------------------------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Задача 4. Прыжки

Ограничение по времени на 1 тест: **1 сек.**

В детстве маленький Ося Бендер был ужасно суеверным и закомплексованным ребенком, а потому старательно перепрыгивал все трещины в асфальте на прямолинейной дорожке. При этом прыжки у него были одной и той же длины, а потому далеко не всегда у него это получалось. Но он был очень упорен и очень тщательно подбирал длину прыжка и точку старта, и вот однажды — СВЕРШИЛОСЬ!!! Он наконец-то перепрыгнул все трещины в асфальте и навсегда запомнил этот день.

При этом он также запомнил и длину прыжка, при которой ему это удалось проделать. После этого его заинтересовал вопрос: "А можно ли это проделать при другой длине прыжка, а если можно, то сколькими способами?" Заметим, что он всегда прыгает только по целочисленным точкам на прямой. Два способа перепрыгнуть все трещины он считает различными тогда и только тогда, когда у них различны длины прыжков — стартовая точка первого прыжка его не интересует.

Ширина трещины пренебрежимо мала по сравнению с расстоянием между точками и длиной прыжка, поэтому сами трещины мы можем считать точками на целочисленной прямой, для которых нам известно их положение — расстояние, отсчитываемое от дома Оси. Считается, что будущему великому комбинатору удалось перепрыгнуть трещины в асфальте, расположенные в точках a_i , если для всех b_i — точек, куда он наступал, выполняется условие $b_{i-1} \leq a_i \leq b_i$ ($1 \leq i \leq N$).

Входные данные

В первой строке входного файла записано два целых числа N и d — количество трещин в асфальте и длина прыжка, при которой Остапу удалось перепрыгнуть все трещины ($1 \leq N \leq 100000$, $0 < d \leq 10^{13}$). В следующей строке записано через пробел N целых чисел a_i — положение трещин в асфальте, заданных в порядке возрастания a_i ($0 \leq a_i \leq 10^{13}$, $1 \leq i \leq N$).

Выходные данные

В выходной файл необходимо вывести одно целое число — количество различных способов перепрыгнуть все трещины в асфальте. Если таких способов бесконечно много, то вывести **-1**.

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|------------------------|-------------------|
| 6 5 3 8 12 18 22 28 | 2 |

Комментарий

Существуют две различные длины прыжка — 5 и 6.

Задача 5. Рога и копыта

Ограничение по времени на 1 тест:

3 сек.

Секретарши конторы «Рога и копыта» очень любили следующее времяпровождение — из имеющихся в первой комнате a_1 рогов и b_1 копыт они случайным образом выбирали c_1 предметов и переносили во вторую. Далее процедура повторялась — из имеющихся во второй комнате $a_2+b_2+c_1$ предметов (a_2 имевшихся рогов, b_2 имевшихся копыт и c_1 добавленных из первой комнаты предметов, среди которых могли быть и рога и копыта) случайным образом выбирается c_2 предметов, которые переносятся в третью комнату и так далее.

В сей замечательный процесс решили вмешаться Паниковский и Балаганов. С одной стороны, они очень не любят рога (а кто же их любит?) и потому стремятся минимизировать мат. ожидание количества выносимых из последней комнаты рогов. С другой стороны, они обнаружили, что за $p \cdot k_i$ шоколадок секретарша из i -ой комнаты готова смухлевать и добавить p копыт к общему набору рогов и копыт в этой комнате (еще до процесса переукладывания) — при этом в этой комнате будет не a_i рогов и b_i копыт, а, соответственно, a_i рогов и b_i+p копыт.

Итак, ваша задача — распределить имеющиеся шоколадки так, чтобы после всех переукладываний мат. ожидание количества рогов, случайным образом выбранных из последней комнаты, было бы минимальным.

Входные данные

В первой строке входного файла записано два целых числа N и M — количество комнат и количество имеющихся шоколадок ($1 \leq N \leq 500$, $0 \leq M \leq 500$).

В каждой из следующих N строк записано через пробел по четыре целых числа a_i , b_i , c_i и k_i , характеризующих каждую комнату — количество имеющихся там первоначально рогов, количество имеющихся там первоначально копыт, количество предметов, случайно выбираемых и переносимых в следующую комнату и взяточность i -ой комнаты, т.е. сколько необходимо затратить шоколадок для добавления одного копыта. Для последней комнаты указывается c_N — количество предметов, среди которых мы и стремимся минимизировать количество имеющихся рогов ($0 \leq a_i \leq 1000$, $0 \leq b_i \leq 1000$, $0 \leq c_i \leq a_i + b_i + c_{i-1}$, $c_0=0$, $0 \leq k_i \leq 1000$).

Входные данные

В выходной файл необходимо вывести одно вещественное число — минимально возможное мат. ожидание количества рогов среди c_N вынесенных из последней комнаты предметов с абсолютной или относительной погрешностью не более 10^{-8} .

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|---------------------------|-------------------|
| 2 5 2 2 3 1 3 4 5 2 | 1.75000000 |

Пояснение

Математическим ожиданием случайной величины, принимающей значения x_i с вероятностями p_i ($1 \leq i \leq N$) называется величина, равная $\sum_{i=1}^N p_i \cdot x_i$.

Задача 6. Бендер на качалке

Ограничение по времени на 1 тест: **3 сек.**

После переквалификации в управдомы и последующего выхода на пенсию Остап Ибрагимович наконец-то осуществил свою мечту — купил кресло-качалку и поставил его внутри комнаты, которая имеет форму многоугольника (не обязательно выпуклого). Посидел он в этом кресле годик-другой, да и решил, что пора бы уж в нем и раскачиваться начинать.

Но перед тем как начать раскачивание, он хочет повернуть кресло так, чтобы обозреваемая им площадь (внутри сектора с фиксированным углом) из этой точки была бы как можно больше. И пока он не повернет кресло в нужном направлении, и не вычислит, какую площадь он при этом увидит, раскачиваться он не начнет.

Так давайте поможем великому комбинатору на пенсии, вычислим за него максимальную возможную площадь внутри многоугольника, которую он может увидеть внутри сектора с углом a из фиксированной точки.

Входные данные

В первой строке входного файла записано три числа x_0 , y_0 , a — соответственно координаты точки, где находится кресло и обозреваемый им угол в градусах ($0 < a < 180$). Во второй строке записано число N — количество вершин многоугольника ($3 \leq N \leq 100000$). В следующих N строках приведены координаты вершин многоугольника по два числа в строке, записанных через пробел, в порядке их обхода против часовой стрелки.

Все числа во входном файле целые. Все координаты по модулю не превосходят 20000. Гарантируется, что многоугольник не имеет самопересечений, и точка, где находится качалка, находится строго внутри него.

Выходные данные

В выходной файл необходимо вывести одно вещественное число — максимально возможную площадь внутри сектора с абсолютной или относительной погрешностью не более 10^{-7} .

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|--|-------------------|
| 1 1 60 5 0 0 1 0 2 0 2 2 0 2 | 0.732050807569 |

Задача 7. Волейбол

Ограничение по времени на 1 тест: **1 сек.**

В правилах игры в волейбол есть следующие пункты:

8.3 Мяч "в площадке"

Мяч считается *в площадке*, когда он касается поверхности игровой площадки, включая ограничительные линии.

8.4 Мяч "за"

Мяч считается *за*, когда:

8.4.1 часть мяча, которая касается пола, находится полностью за ограничительными линиями.

Задача состоит в том, чтобы определять, находится ли мяч *в площадке* или *за*. Частью мяча, которая касается пола, можно для простоты считать круг (мяч вдавливается при ударе о площадку).

Входные данные

В первой строке входного файла записано число N — количество запросов ($1 \leq N \leq 10000$). Далее следует N строк, содержащих по одному запросу в каждой. Запрос состоит из пяти вещественных чисел, заданных с точностью до пяти знаков после десятичной точки.

Первые два числа l и w — длина и ширина площадки вместе с ограничительными линиями. Длина l отсчитывается по оси OX , ширина w — по оси OY . Левый нижний угол площадки, включая ограничительные линии, находится в точке $(0; 0)$.

Далее следуют три числа x , y , r — координаты точки центра касания пола мячом и радиус круга касания ($0.00001 \leq l, w, r \leq 10000$; $-10000 \leq x, y \leq 10000$).

Выходные данные

Выходной файл должен содержать N строк, в которых для каждого запроса нужно вывести одно из слов: **In**, если мяч *в площадке*, и **Out**, если мяч *за*.

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|---|-------------------|
| 2 | In |
| 1.00000 0.50000 0.80000 0.30000 0.10000 | Out |
| 1.00000 0.50000 2.00000 0.30000 0.10000 | |

Задача 8. Мнение хоббитов

Ограничение по времени на 1 тест: **1 сек.**

Как известно, Бильбо Бэггинс живёт в Шире. Всего там живёт не более N хоббитов. О каждом событии в Шире каждый хоббит сразу имеет своё предварительное мнение, выраженное числами -1 , 0 или 1 . Однако каждый хоббит обязательно корректирует своё мнение в соответствии с мнением тех своих знакомых, которые свое мнение ему сообщают. Если с хоббитом никто своим мнением не делится, то его мнение не изменяется, так как не с кем его корректировать. В противном случае этот процесс состоит из трёх этапов:

1 этап: хоббит принимает *предварительное мнение*;

2 этап: (*повторяется M суток подряд*)

- если мнение у хоббита уже есть, то он в течение дня собирает мнения своих знакомых, а вечером определяет *мнение знакомых* по следующей формуле:

$$\frac{1}{|A|} \sum_{k \in A} \text{мнение}(k) \times \text{доверие}(k),$$

где A — множество всех знакомых данного хоббита, которые сообщают свои мнения данному хоббиту, $|A|$ — количество таких знакомых, $\text{мнение}(k)$ — мнение хоббита с номером k , $\text{доверие}(k)$ — степень доверия к хоббиту с номером k , которая измеряется теми же целыми числами -1 , 0 и 1 ;

- ночью хоббит определяет своё *новое мнение* как

$$\frac{1}{2} (\text{прежнее мнение} + \text{мнение знакомых});$$

3 этап: утром $M+1$ дня хоббит округляет полученное значение до целого по правилу:

$$[-1, -0.5] \rightarrow -1, \quad (-0.5, 0.5) \rightarrow 0, \quad [0.5, 1] \rightarrow 1.$$

Обычно дело ограничивается одними сутками ($M = 1$), но для ОСОБО ВАЖНЫХ СОБЫТИЙ процесс затягивается на 7 недель ($M = 49$)!

В некоторый день произошло ОСОБО ВАЖНОЕ СОБЫТИЕ — в Шир приехал волшебник Гэндальф, и у всех хоббитов сразу появилось (и оно известно вам) предварительное мнение об этом событии. Известны личные связи хоббитов (кто кому сообщает свое мнение) и кто как доверяет мнению знакомых. Спрашивается, какое окончательное мнение сложится у всех хоббитов о приходе Гэндальфа?

Входные данные

В первой строке входного файла записано целое число N — количество хоббитов ($1 \leq N \leq 100$). Все хоббиты пронумерованы числами от 1 до N .

Во второй строке записано через пробел N чисел — предварительные мнения хоббитов в порядке их нумерации.

Далее для каждого хоббита указывается информация о тех его знакомых, кто сообщает ему свое мнение. Она состоит из трёх строк. В первой строке записывается количество таких знакомых хоббита. Во второй строке перечисляются номера этих знакомых данного хоббита в произвольном порядке, а в третьей — степень его доверия к каждому из них соответственно.

Выходные данные

Выходной файл должен содержать N строк, в которых записано окончательное мнение хоббитов в порядке их нумерации — по одному числу в строке.

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|------------------|-------------------|
| 2 | 0 |
| 1 0 | 0 |
| 1 | |
| 2 | |
| 1 | |
| 1 | |
| 1 | |

0

Задача 9. Сломанный артефакт

Ограничение по времени на 1 тест: **3 сек.**

Давным-давно, во вселенной одной ролевой игры существовал могущественный артефакт. Чтобы избежать его случайного использования, артефакт был разделен на две части и спрятан на разных концах одного континента. Зная, что части были расположены так, что кратчайший путь между ними по суше максимален, игрок может догадаться, где именно они спрятаны.

Входные данные

В первой строке входного файла даны два целых числа N и M – размеры карты континента ($1 \leq N \leq 100$, $1 \leq M \leq 100$). В следующих N строках дано по M символов '#' и '-', где '#' обозначает клетки, занятые континентом, а символ '-' обозначает клетки, занятые морем. Игрок может перемещаться только по суше вверх, вниз, влево и вправо. Гарантируется, что из любой точки суши можно дойти до любой другой точки суши. Гарантируется, что имеются, по крайней мере, две различные клетки суши.

Выходные данные

Выходной файл должен содержать N строк по M символов '#', '-' и 'X' — исходная карта, где клетки, в которых расположен артефакт, отмечены 'X'. Если вариантов расположения артефакта несколько, то вывести любой.

Пример

| <i>input.txt</i> | <i>output.txt</i> |
|---|----------------------------------|
| 4 5 -###- ##### #-### ##--- | -###- ##### #-##X #X--- |

Задача 10. Пентюшкин за границей

Ограничение по времени на 1 тест: **2 сек.**

Пентюшкин поехал по обмену учиться в Европу. На выходные он хочет съездить к друзьям в другой город. Для этого он взял напрокат машину. Ему необходимо определить оптимальное расписание заправок бензином, такое, чтобы доехать до своей цели с минимальными затратами.

Входные данные

В первой строке входного файла заданы четыре числа S , M , V и N , разделенные пробелом, где S — это расстояние в километрах по дороге между двумя городами, заданное с точностью до третьего знака после десятичной точки, M — количество горючего в литрах, которое машина Пентюшкина потребляет за один километр, заданное не более чем с двумя знаками после десятичной точки, V — целое число, ёмкость бака машины в литрах, N — целое число, количество заправочных станций на дороге ($100 \leq S \leq 2000$, $0.01 \leq M \leq 10$, $10 \leq V \leq 50$, $1 \leq N \leq 100$).

В следующих N строках для каждой из заправочных станций указано два числа S_i и C_i , где S_i — расстояние по дороге от города, где учится Пентюшкин, до нее в километрах, заданное с точностью до третьего знака после десятичной точки, а C_i — цена бензина на этой станции в евро за литр с точностью до евроцента (одной сотой евро) за литр ($0 \leq S_i \leq S$, $0.01 \leq C_i \leq 200$, $1 \leq i \leq N$). Ближайшая заправочная станция находится на расстоянии 0.000 километров от города. Гарантируется, что поездка всегда возможна. Счётчики бензина на заправочных станциях позволяют отмерять сотые доли литра. Некратное сотой доле литра количество бензина купить нельзя. Запрещается выливать лишний бензин. Цена заправки округляется вверх до целого числа евроцентов.

Выходные данные

В первую строку выходного файла необходимо вывести минимально возможную стоимость бензина, требуемого для поездки Пентюшкина, в евро ровно с двумя знаками после десятичной точки.

В следующей строке должно быть записано целое число K — количество заправочных станций, на которых ему придётся заправляться. Далее в K строках для каждой заправочной станции в порядке их прохождения нужно вывести через пробел номер станции и количество бензина, купленного на ней. Количество бензина выдавать ровно с двумя знаками после десятичной точки.

Примеры

| <i>input.txt</i> | <i>output.txt</i> |
|--|---|
| 100 0.1 10 2 0 50 50 20 | 350.00 2 1 5.00 2 5.00 |
| 100 0.12 10 3 78 10.91 0 15.45 33 12.34 | 156.64 3 2 3.96 3 5.40 1 2.64 |