

Задача А. Держать строй - 3

Имя входного файла: `army3.in`
Имя выходного файла: `army3.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В воинской части города Шатров продолжают занятия по строевой подготовке. На этот раз Андрей Юрьевич выполняет очередное задание своего начальника Павла Андреевича. Для выполнения этого задания, Андрею Юрьевичу необходимо среди всех n солдат, стоящих в одной шеренге, выбрать отряд из k высоких солдат для выполнения строительных работ.

В качестве первого шага, Андрей Юрьевич приказал каждому солдату посчитать свой *показатель роста*. Для этого каждый солдат, стоящий в шеренге, должен посмотреть сначала в одну сторону и посчитать количество солдат в этой части шеренги, которые строго ниже его, потом посмотреть в другую сторону, посчитать такое же количество, и тогда сумма этих двух чисел и будет его *показателем роста*.

На втором шаге, основываясь на этом показателе, Андрей Юрьевич должен выбрать отряд. Поскольку за долгие дни и ночи занятий строевой подготовкой солдаты успели хорошо познакомиться и даже подружиться со своими соседями в шеренге, Андрей Юрьевич решил выбрать в шеренге такой непрерывный подотрезок из ровно k солдат, у которого сумма показателей роста всех солдат максимальна.

Обладая информацией о росте каждого солдата в шеренге, помогите Андрею Юрьевичу найти оптимальный подотрезок.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и k ($1 \leq k \leq n \leq 100000$) — количество солдат в строю и необходимый размер отряда, соответственно. Следующая строка содержит n целых чисел h_i ($1 \leq h_i \leq 10^9$) — рост i -го слева солдата в шеренге.

Формат выходного файла

В выходной файл выведите одно целое число l — левый конец подотрезка из k солдат с максимальным показателем роста. Если таких отрезков несколько, выведите самый левый.

Примеры

| <code>army3.in</code> | <code>army3.out</code> |
|-----------------------|------------------------|
| 4 2 1 2 4 3 | 3 |
| 4 2 2 1 1 2 | 1 |

Задача В. Шахматная доска

Имя входного файла: `chess.in`
Имя выходного файла: `chess.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася любит играть в необычные шахматы. Как-то раз, когда его братишка Коля был еще очень маленький, Вася вернулся из школы, и увидел, что его любимую шахматную доску кто-то перекрасил. Вася не сильно разозлился на Колю, потому что очень любил своего младшего брата. Так как у них дома были только черная и белая краски, каждая клетка доски была покрашена в один из этих двух цветов.

Вася решил исправить ошибку брата, он решил покрасить доску так, чтобы она снова стала шахматной. Но Вася почему-то подумал, что хочет красить только диагонали, причем сразу целиком. Вася решил не тратить много времени, поэтому его интересует способ покраски, который содержит наименьшее количество действий. За одно действие Вася может покрасить полностью какую-либо диагональ, в любой из двух цветов: черный или белый. Диагонали бывают двух типов, в зависимости от направления прямой, на которой лежит диагональ. Диагональ, которая лежит на прямой, направленной влево и вниз, является диагональю первого типа, а диагональ, которая лежит на прямой, направленной вправо и вниз, — второго.

Вам предстоит помочь Васе. Задано испорченное Колей шахматное поле. Вам необходимо определить, за какое минимальное количество действий Вася сможет перекрасить доску так, чтобы она стала шахматной.

Формат входного файла

В первой строке входного файла записаны два целых числа: n и m ($1 \leq n, m \leq 100$) — количество строчек и количество столбцов шахматного поля, соответственно.

В следующих n строках записано поле. Каждая строка содержит по m символов, и описывает одну строку шахматного поля. Символ `W` соответствует белой клетке, `B` — черной.

Формат выходного файла

В первой строке выходного файла выведите число p — минимальное количество действий, которое потребуется Васе, чтобы его доска снова стала шахматной.

В следующих p строках выведите описания действий. Каждое действие описывается тремя параметрами: тип диагонали, координаты некоторой покрашенной клетки и цвет. Тип диагонали — это число 1 или 2 — тип диагонали. Координаты клетки — это два целых числа: строка и столбец одной из клеток, которую покрасили этим действием. Цвет — это символ `W` или `B`, белый и черный соответственно.

Примеры

| <code>chess.in</code> | <code>chess.out</code> |
|--------------------------|------------------------|
| 3 3 WBB BWB BBW | 1 1 1 3 W |
| 3 3 WBW WWB WWW | 1 2 2 1 B |
| 1 3 WWW | 1 1 1 2 B |

Задача С. Строки Фибоначчи

Имя входного файла: `fibstring.in`
Имя выходного файла: `fibstring.out`
Ограничение по времени: 2 секунд
Ограничение по памяти: 256 мегабайт

Алла очень любит палиндромы. Все потому, что её имя является палиндромом. Напомним, что строку называют палиндромом тогда, когда она одинаково читается как слева направо, так и справа налево.

Однажды в школе учитель рассказал Алле про так называемые строки Фибоначчи. Строки Фибоначчи определяются следующим образом:

- $f_0 = a$
- $f_1 = b$
- $f_n = f_{n-1}f_{n-2}$ для каждого $n \geq 2$ — конкатенация двух предыдущих строк Фибоначчи

Таким образом, первые пять строк Фибоначчи: «a», «b», «ba», «bab», «babba».

Аллу сразу заинтересовал вопрос — какой максимально длинный палиндром встречается в k -й строке Фибоначчи. Помогите Алле решить эту задачу.

Формат входного файла

Первая строка входного файла содержит одно целое число k ($0 \leq k \leq 80$) — номер строки Фибоначчи.

Формат выходного файла

В выходной файл выведите длину самого большого палиндрома, содержащегося в k -й строке Фибоначчи.

Примеры

| <code>fibstring.in</code> | <code>fibstring.out</code> |
|---------------------------|----------------------------|
| 2 | 1 |
| 4 | 4 |

Задача D. Перелет

Имя входного файла: `flight.in`
Имя выходного файла: `flight.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Далеко не во всех больших странах хорошо развита внутренняя авиация. Так, иногда, чтобы попасть из одного большого города в другой, приходится делать это двумя рейсами с пересадкой в столице. Зато в таких странах обычно не возникает проблем с попаданием из любого города в столицу и из столицы куда угодно.

Итак, вы разрабатываете систему бронирования авиабилетов. Конкретно сейчас вам необходимо реализовать ту ее часть, которая по списку рейсов из города А в столицу страны и списку рейсов из столицы в город В выведет самый дешевый способ попасть из города А в город В. При этом важно, что на стыковку должно быть заложено не меньше 15 минут. То есть, если пассажир приземлился в столице в 23:00:00, то на рейс в В в 23:15:00 он успеет, а на рейс в 23:14:59 — нет.

В случае, если существует несколько вариантов с одинаковой суммарной ценой, и эта цена минимальна, выведите тот вариант, в котором меньше время прибытия в В. Если существует несколько таких вариантов — выведите любой из них.

Формат входного файла

Первая строка входного файла содержит одно натуральное число n ($1 \leq n \leq 100\,000$) — количество рейсов из города А в столицу государства. В следующих n строках описаны эти рейсы. Описание рейса состоит из его номера, времени вылета, времени прилета и цены билета на этот рейс, разделенных пробелами. Номер рейса — набор заглавных латинских букв и цифр длиной не более пяти символов. Времена описаны в формате `hh:mm:ss`. Гарантируется, что для любого рейса время прибытия больше, чем время вылета. Цена билета — натуральное число, не превышающее 100 000.

Следующая строка входного файла содержит одно натуральное число m ($1 \leq m \leq 100\,000$) — количество рейсов из столицы в город В. В следующих m строках описаны рейсы в В в формате, аналогичном описанному выше.

Гарантируется, что не существует двух рейсов с одинаковыми номерами.

Формат выходного файла

В первой строке выведите номер рейса, которым пассажир полетит в столицу. Во второй — номер рейса, которым он полетит из столицы в город В. При этом в городе В пассажир должен оказаться в те же сутки, в которые он вылетел из города А, то есть переход через сутки во время ожидания стыковки невозможен.

Примеры

| <code>flight.in</code> | <code>flight.out</code> |
|------------------------------|-------------------------|
| 3 | UT135 |
| UT135 08:00:00 11:00:00 1350 | PK074 |
| FV42 09:00:00 10:50:00 1500 | |
| A123B 12:00:15 16:50:38 2000 | |
| 3 | |
| PK074 17:30:00 23:00:00 1350 | |
| NV074 11:04:59 22:59:00 1100 | |
| AK59 10:00:00 19:00:00 200 | |

Задача Е. Кубок Гагарина

Имя входного файла: `gagarincup.in`
Имя выходного файла: `gagarincup.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Хоккей с шайбой — один из самых распространенных в России видов спорта. На днях закончился розыгрыш самого престижного в Европе хоккейного клубного турнира — Кубка Гагарина.

На первом этапе плей-офф борьбу начинает 16 клубов, разбитые по парам. Далее команды, попавшие в одну пару играют между собой серию из семи матчей до четырех побед. Если одна из команд выигрывает четыре матча, то серия прекращается и она признается победителем. Далее остается восемь команд, которые играют второй раунд по тем же правилам, и так далее, вплоть до победителя.

Первые два матча серии проходят на площадке первой команды, следующие два на площадке второй команды, после этого следующие матчи (если они нужны) проходят поочередно сначала на площадке первой команды, потом второй, то есть по схеме 2-2-1-1-1.

Вам дана вероятность победы каждой из команд на каждой из площадок. Вам нужно определить вероятность, с которой серия завершится именно с данным счетом.

Формат входного файла

Первая строка входного файла содержит вещественных два числа a и b — вероятность победы каждой из команд на площадке первой команды ($0 \leq a, b \leq 1, a+b = 1$), вторая строка в аналогичном формате вероятность побед из команд на площадке второй команды. Третья строка содержит счет, вероятность которого Вам нужно определить.

Формат выходного файла

Выведите одно число — ответ на задачу. Ответ должен отличаться от правильного не более, чем на 10^{-6} .

Примеры

| <code>gagarincup.in</code> | <code>gagarincup.out</code> |
|----------------------------|-----------------------------|
| 0.7 0.3 | 0.142884 |
| 0.54 0.46 | |
| 4-0 | |

Задача F. Волшебный ящик

Имя входного файла: magicbox.in
Имя выходного файла: magicbox.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Владислав очень любит различные шоу иллюзионистов. После просмотра очередного шоу его очень заинтересовал номер, в котором в ящик залезала помощница иллюзиониста и он протыкал этот ящик множеством шпаг, после чего девушка живая и невредимая вылезала обратно.

Владислав решил дома попробовать повторить этот фокус. В качестве ящика он взял прямоугольный параллелепипед размером $L \times D \times D$. Поскольку Владиславу не хотелось подвергать опасности человеческую жизнь, он решил поместить в ящик свой любимый футбольный мяч, который имеет форму шара с диаметром ровно D .

Помимо этого, юный фокусник разметил на сторонах параллелепипеда, имеющих размер $D \times L$, несколько точек, в которых он будет протыкать ящик шпагами насквозь. Грани квадратной формы (размером $D \times D$) он решил не протыкать.

Во время выполнения фокуса Владиславу все не удавалось разместить мяч так, чтобы не было опасности проткнуть его одной из шпаг. Поскольку Владислав очень дорожит своим мячом, он просит вас выяснить, существует ли такое расположение мяча в ящике, при котором ни одна шпага не проткнет его (при этом, возможно, будет касаться его). Каждая шпага при этом считается пренебрежимо тонкой.

Формат входного файла

Первая строка входного файла содержит три целых числа: N , D и L ($0 \leq N \leq 100000$, $1 \leq D, L \leq 10^9$) — количество шпаг и размеры ящика и мяча.

Следующие N строк содержат описания точек, в которых ящик будет проткнут шпагами.

Для удобства описания этих мест, Владислав расположил ящик так, что прямо перед ним расположена одна из сторон, размером $D \times L$, которую он назвал передней и ввел на ней прямоугольную систему координат с центром в левом нижнем углу. Правый верхний угол при этом имеет координату (L, D) . Аналогичную систему координат он ввел на верхней грани, которая также имеет размер $D \times L$, при этом смотрел на нее он сверху. Поскольку Владислав не протыкает ящик шпагами со стороны квадратной грани (размером $D \times D$), то этих систем координат ему хватило для обозначения всех точек протыкания.

Строка вида $F \ x \ y$ ($0 \leq x \leq L$, $0 \leq y \leq D$) описывает протыкание передней грани в точке с координатами (x, y) . А строка вида $U \ x \ y$ ($0 \leq x \leq L$, $0 \leq y \leq D$) описывает протыкание верхней грани в точке с координатами (x, y) . Все координаты целые. Все точки различны.

Формат выходного файла

В выходной файл выведите «YES», если возможно так расположить мяч в ящике, чтобы он не был проткнут ни одной шпагой, или «NO» в противном случае.

Примеры

| magicbox.in | magicbox.out |
|-------------------------|--------------|
| 2 2 4 U 1 1 F 3 1 | YES |
| 2 2 3 U 1 1 F 2 1 | NO |

Задача G. Благовонное число

Имя входного файла: `number.in`
Имя выходного файла: `number.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Марата замечательное обаяние, и он даже различает запах чисел. Это необычайно редкий дар. Его внуки очень ему завидуют, и постоянно спрашивают у него, как пахнет то или иное число.

Все числа пахнут по-разному, но некоторые пахнут особенно приятно. Именно их больше всего любит дедушка Марат. Внуки по очереди давали ему карточки с числами, начиная с единицы, а Марат определял их запах. Внуки отмечали те числа, которые по мнению их дедушки пахнут особенно приятно. Вскоре один из внуков заметил, что наиболее благовонные числа читаются справа налево и слева направо одинаково.

Определите, какое благовонное число внуки отметят N -ым.

Формат входного файла

Первая и единственная строка входного файла содержит одно целое число N , длина которого не превосходит 10^4 цифр.

Формат выходного файла

Выведите одно число — ответ на задачу.

Примеры

| <code>number.in</code> | <code>number.out</code> |
|------------------------|-------------------------|
| 10 | 11 |

Задача Н. Фотография

Имя входного файла: `photo.in`
Имя выходного файла: `photo.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как-то раз, рассматривая фотографии из семейного фотоальбома, Валера заметил одну немаловажную деталь, на которую ранее внимания он не обращал. На одной из фотографий на заднем плане он увидел календарь. Однако, различные предметы заслоняли календарь, да и фотография уже порядком потрепалась, поэтому, полностью рассмотреть дату снимка не представлялось возможным.

А именно, некоторые цифры даты не были видны. Но Валера знает, когда примерно происходило запечатлённое на фотографии. Помогите ему определить, в сколько различных дней могла быть сделана фотография.

Формат входного файла

В трёх строках входного файла, каждая в своей строке, задано три даты: дата, которую увидел Валера, самое раннее, когда это могло произойти, и самое позднее, когда это могло произойти, соответственно. Все даты заданы в формате DD.MM.YYYY. И года, и дни, и месяца, заданы с ведущими нулями, если это необходимо.

В первой дате на любом месте вместо D, M или Y может стоять ?. Это означает, что на фотографии соответствующую цифру рассмотреть невозможно. Все даты находятся в интервале от первого января 1970 года до 19 января 2038 года, включительно. Обратите внимание, что в этот интервал попадают високосные года.

Формат выходного файла

В единственной строке выходного файла выведите единственное целое число — количество дней, в которые могла быть сделана фотография.

Примеры

| <code>photo.in</code> | <code>photo.out</code> |
|--|------------------------|
| ?1.?3.2000 01.03.2000 31.03.2000 | 4 |

Задача I. СМС

Имя входного файла: `sms.in`
Имя выходного файла: `sms.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Одним из самых популярных способов использования мобильной связи являются СМС-сообщения. Каждый день миллионы людей отправляют десятки миллионов сообщений: «Привет!», «Как дела?», «Ты где?», «Я задержусь» и еще тысячи различных коротких посланий.

К сожалению, на клавиатуре мобильного телефона может оказаться меньше кнопок, чем букв в алфавите. Поэтому на первой кнопке размещаются несколько первых букв алфавита, на второй — следующие несколько и так далее. Чтобы набрать некую букву, необходимо нажать ту кнопку, на которой она размещена, t раз, если буква является t -ой по алфавиту, размещенной на этой кнопке. Так, если на первой кнопке клавиатуры размещены буквы a , b и c , то для выбора буквы c придется нажать эту кнопку трижды.

Ученые изучили среднее количество раз, которое каждая буква алфавита встречается в СМС-ках за время службы среднестатистического телефона. Теперь у фирмы-производителя появилась возможность спроектировать клавиатуру так, чтобы минимизировать суммарное количество нажатий на все кнопки. Помогите им сделать это.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \leq n \leq 30$, $1 \leq m \leq 10$) — количество букв в алфавите и кнопок на клавиатуре телефона.

Следующая строка содержит n целых чисел a_i ($0 \leq a_i \leq 1\,000\,000$) — количество раз, которые будет напечатана i -ая буква алфавита.

Формат выходного файла

В выходной файл выведите m целых неотрицательных чисел b_i — количество букв, отнесенных к i -ой кнопке клавиатуры. Сумма всех b_i должна быть равна n .

Если возможных ответов несколько — выведите любой.

Примеры

| <code>sms.in</code> | <code>sms.out</code> |
|---------------------|----------------------|
| 5 3 1 3 5 1 1 | 1 1 3 |

Примечание

В примере на первой кнопке размещена одна первая буква алфавита, на второй — только вторая, а на третьей — оставшиеся три.

Таким образом, суммарное количество нажатий на кнопки будет:
 $1 \times 1 + 3 \times 1 + 5 \times 1 + 1 \times 2 + 1 \times 3 = 14$.

Задача J. Сумма квадратов

Имя входного файла: `sumsquare.in`
Имя выходного файла: `sumsquare.out`
Ограничение по времени: 2 секунд
Ограничение по памяти: 256 мегабайт

Паша и Никита играют в новую компьютерную игру. В игре есть n карточек с числами от 0 до $n - 1$. Карточка с числом b увеличивает силу персонажа на b единиц и увеличивает запас энергии на b^2 единиц. Паша и Никита играют друг против друга. И при этом хотят, чтобы игра была как можно более интересной. Для этого они решили, что их персонажи должны иметь одинаковую силу и одинаковый запас энергии. Помогите им.

Более формально, у вас есть набор чисел от 0 до $n - 1$. Вам требуется его разбить на два таких непересекающихся набора a_i и b_j , таких что $\sum a_i = \sum b_j$ и $\sum a_i^2 = \sum b_j^2$.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 100\,000$) — количество карточек.

Формат выходного файла

В первой строке выведите «No», если невозможно разбить на два таких набора, или выведите «Yes», если возможно. Если это возможно, во второй строке выведите числа, принадлежащие одному из двух наборов, разделенные пробелами. Числа можно выводить в любом порядке.

Примеры

| <code>sumsquare.in</code> | <code>sumsquare.out</code> |
|---------------------------|----------------------------|
| 8 | Yes 0 3 5 6 |
| 2 | No |

Примечание

В первом примере:

$$0 + 3 + 5 + 6 = 1 + 2 + 4 + 7 = 14$$

$$0^2 + 3^2 + 5^2 + 6^2 = 1^2 + 2^2 + 4^2 + 7^2 = 70$$