

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. Выражение

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

У Ивана было n карточек, на каждой из которых написано одно целое число от -100 до 100 . Он часто переставлял их, и однажды он заметил, что он расставил их так, что сумма первых $n - 1$ чисел равна последнему числу, то есть $\sum_{i=1}^{n-1} a_i = a_n$. Но пока Иван обедал, его младшая сестра Алёна перемешала все карточки. К сожалению, Иван не запомнил число, которое являлось суммой остальных чисел. Помогите Ивану найти его.

Формат входного файла

Первая строка содержит одно целое число n ($2 \leq n \leq 100$) — количество карточек. Вторая строка содержит n целых чисел a_i ($|a_i| \leq 100$) — числа написанные на карточках.

Формат выходного файла

Выведите минимальный номер карточки j такой, что a_j равно сумме чисел на всех остальных карточках.

Примеры

<code>expression.in</code>	<code>expression.out</code>
3 1 3 2	2

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».

Алле 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. 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 начинают 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 Е. Игра с графом

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

В одном из очень известных графств завелись два азартных игрока: Алиса и Боб. Граф очень встревожен этим, потому что от таких игроков страдают люди и его графство. Вот и на этот раз Алиса и Боб придумали новую игру.

Опишем правила игры. Игроки делают ходы по очереди. На каждом ходе игрок выбирает дорогу, которая еще не разрушена, и разрушает ее. Если после хода игрока, граф видит, что из какого-то города нельзя добраться во все остальные по неразрушенным дорогам, то он выгоняет этого игрока из графства за порчу имущества, и этот игрок проигрывает в игре своему оппоненту.

Известно, что графство состоит из городов и дорог, причем каждая дорога соединяет ровно два города. По дорогам можно перемещаться в обоих направлениях. Не существует дорог, которые соединяет город сам с собой, и каждую пару городов соединяет не более одной дороги. Из каждого города можно добраться во все другие города.

Известно, что каждый из игроков хочет выиграть, поэтому действует всегда оптимально.

Вам задано графство и известно, что Боб, как джентельмен, уступает первый ход Алисе. Вам нужно понять, кто проиграет в данной игре.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m ($1 \leq n \leq 100$, $0 \leq m \leq 500$) — количество городов и дорог соответственно. В следующих m строках заданы дороги — два числа: a и b ($1 \leq a < b \leq n$) — номера городов, которые соединяет соответствующая дорога.

Гарантируется, что никакая пара городов не встречается во входном файле дважды и что из любого города можно добраться до всех остальных по заданным дорогам.

Формат выходного файла

В выходной файл выведите имя игрока, который проиграл: Alice или Bob, либо Draw, если никто не проиграл.

Примеры

<code>graphgame.in</code>	<code>graphgame.out</code>
3 3 1 2 1 3 2 3	Bob
2 1 1 2	Alice

Примечание

В первом примере Алиса разрушает любую из трех дорог. После этого Боб, вне зависимости от своего хода, он проигрывает.

Во втором примере всего одна дорога, разрушив ее, Алиса проиграет.

Задача F. Монстры

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

В новой компьютерной игре для прохождения 85-го уровня игроку требуется уничтожить монстров в k комнатах. В каждой комнате изначально находится a_i монстров, и единственное, что может делать игрок — создавать существ по имени Февроний.

Каждое созданное существо по имени Февроний голодно и хочет насытиться, поедая монстров. Однако, с каждым ходом сила игрока растет, и поэтому Февронию первому необходим ровно один монстр, Февронию второму — два, третьему — четыре, i -му — 2^{i-1} .

После создания очередного Феврония игрок указывает ему на одну из k комнат, после чего Февроний идет туда. Если в этой комнате достаточно монстров для его насыщения, то он съедает столько монстров, сколько ему нужно, и умирает со счастливой улыбкой на лице. Если же ему не хватает хотя бы одного монстра, то он съедает всех, после чего выходит из комнаты и съедает игрока. Естественно, что такой вариант развития событий крайне нежелателен.

Помогите игроку выяснить, сможет ли он, создавая Феврониев, уничтожить всех монстров и остаться несъеденным.

Формат входного файла

В первой строке входного файла дано одно целое число k ($1 \leq k \leq 5$) — количество комнат. В следующей строке даны k целых чисел a_i ($1 \leq a_i < 1024$) — количества монстров в комнатах.

Формат выходного файла

Выведите в выходной файл «Yes», если игрок пройдет уровень, и «No» — в противном случае.

Примеры

<code>monsters.in</code>	<code>monsters.out</code>
2 21 10	Yes
3 22 10 32	No

Задача G. 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 Н. 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:
 $1 \times 1 + 3 \times 1 + 5 \times 1 + 1 \times 2 + 1 \times 3 = 14$.