

Задача А. Дистанционное обучение

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Пандемия нового компьютерного вируса добралась до Байтландии. Для безопасности байтов (так зовут жителей Байтландии) школам пришлось перейти на дистанционное обучение. Знакомо, не правда ли?

Онлайн-уроки будут проходить в виде конференций, в каждой конференции может находиться максимум n человек, s из них обязательно преподаватели, а остальные ученики. У школы в расписании есть k свободных конференций.

Помогите школе определить, какое максимальное количество учеников может находиться на конференциях, если каждый ученик может принимать участие максимум в одной конференции.

Поспешите, ведь начало уроков совсем скоро!

Формат входных данных

В первой строке записано целое число k ($1 \leq k \leq 1000$) — количество свободных конференций у школы.

Во второй строке записано целое число n ($1 \leq n \leq 100$) — максимальное количество участников конференции.

Во третьей строке записано целое число s ($1 \leq s < n$) — количество преподавателей на конференции.

Формат выходных данных

Выведите единственное число — максимальное количество учеников, которые могут находиться на конференциях.

Примеры

стандартный ввод	стандартный вывод
1 10 1	9
2 4 2	4

Задача В. Код от сейфа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В результате оперативной работы в руки к Штирлицу попало четырёхзначное натуральное число n , являющееся зашифрованным кодом от сейфа Мюллера. Ещё раньше Штирлицу удалось установить, что сейф в кабинете Мюллера открывается с помощью кодового механического замка, который содержит кнопки с цифрами от 0 до 9. Также Штирлиц знал, что Мюллер никогда не пользовался кнопкой 1, так как она была неисправна с момента установки сейфа.

Через некоторое время Штирлицу удалось установить используемую Мюллером методику шифрования. А именно, придумав код, Мюллер выписывал подряд цифры кода, получая при этом натуральное число, а затем умножал полученное число на каждую из его цифр.

Помогите Штирлицу разгадать код от сейфа Мюллера и овладеть секретными документами или определить, что данные, которые попали к Штирлицу, некорректны.

Формат входных данных

Первая строка содержит четырёхзначное натуральное число n .

Формат выходных данных

Необходимо вывести код от сейфа Мюллера.

Если же код нельзя определить однозначно, то вывести сообщение «ERROR».

Примеры

стандартный ввод	стандартный вывод
6111	97
7055	ERROR

Задача С. Всеобъемлющая Галактическая Магистральная Сеть

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Всей Галактике известна Всеобъемлющая Галактическая Магистральная Сеть, соединяющая многие звёздные системы между собой. Эта транспортная сеть объединяет n звёздных систем посредством $n - 1$ магистралей — скоростных путей, соединяющих некоторые две звёздные системы между собой.

Главной особенностью этой магистральной сети является существование между любыми двумя звёздными системами ровно одного маршрута по этой сети.

У Галактической Федерации появилась возможность добавить в сеть одну дополнительную магистраль, соединяющую две разные звёздные системы, между которыми ещё не существует магистрали и, тем самым, разгрузить магистральную сеть.

Сложность постройки такой магистрали вызвано наличием развилок. Чтобы разгрузить транспортную сеть по максимуму, в любой звёздной системе существуют развилки. Развилки соединяют уникальным туннелем каждую пару магистралей, входящих в звёздную систему.

Таким образом, Федерация решила оценить, какое максимальное количество развилок может получиться во Всеобъемлющей Галактической Магистральной Сети после добавления одной магистрали, и поручила эту задачу Вам.

Формат входных данных

В первой строчке записано целое число n ($3 \leq n \leq 2 \cdot 10^5$) — количество звёздных систем в сети.

Далее идёт $n - 1$ строчка, описывающая пары звёздных систем u_i и v_i ($1 \leq u_i, v_i \leq n$), соединённые магистралями.

Формат выходных данных

В первой строчке выведите максимальное количество развилок, которое можно получить после добавления одной магистрали.

Во второй строчке выведите пару звёздных систем, соединив которые можно добиться максимального количества развилок.

Примеры

стандартный ввод	стандартный вывод
4 1 2 2 3 3 4	5 1 3
6 1 2 1 3 1 4 4 5 4 6	10 1 5

Задача D. ДНК-палиндром

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Цепочка ДНК состоит из последовательности повторяющихся блоков, или нуклеотидов. Существуют четыре типа нуклеотидов, которые обозначаются латинскими буквами A, C, G, и T. Таким образом, каждая цепочка может быть схематично представлена как строка над алфавитом из этих четырех символов.

В большинстве случаев у молекулы ДНК есть две цепочки, которые комплиментарны друг другу. Для того, чтобы построить обратно-комплиментарную цепочку ДНК для заданной цепочки s , необходимо заменить нуклеотиды на комплиментарные, и прочитать их в обратном порядке. Комплиментарным нуклеотидом для A является T, для T — A, для C — G и для G соответственно C. Например, обратно комплиментарной цепочкой для ACG будет CGT.

ДНК-палиндромом называется цепочка ДНК, которая совпадает со своей обратно-комплиментарной цепочкой.

Для данной строки ДНК определите, является ли она ДНК-палиндромом.

Формат входных данных

В первой строке находится одно целое число n ($1 \leq n \leq 10^6$) — количество символов в строке ДНК.

Во второй строке содержится n символов из алфавита $\{A, C, G, T\}$, представляющие строку. Все символы находятся в верхнем регистре.

Формат выходных данных

Выведите «YES», если строка является ДНК-палиндромом, и «NO» в обратном случае.

Примеры

стандартный ввод	стандартный вывод
4 ATAT	YES
3 AAA	NO

Задача E. Разлад Империй

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На далёком-далёком континенте расположены n городов. Между ними располагаются m дорог: каждая соединяет два различных города и позволяет перемещаться между ними в обе стороны.

На континенте представлены d империй, которые пронумерованы от 1 до d . Каждая империя имеет в распоряжении хотя бы один город, а каждый город принадлежит одной из империй, а именно, i -й город числится в составе империи e_i .

Каждая империя беспокоится, что ей скоро объявит войну какая-то одна другая империя, поэтому все империи хотят расставить армейские штабы в каких-то своих городах.

Штабы определённой империи должны быть расставлены так, чтобы независимо от того, кто враг, из каждого города этой империи существовал безопасный путь в какой-либо штаб этой империи. Путь считает безопасным, если в нём отсутствуют города, принадлежащие вражеской империи.

Каждая империя хочет расставить минимальное количество таких штабов. Помогите континенту — определите штабы для каждой империи.

Формат входных данных

Первая строка содержит три целых числа n , m и d — количество городов, дорог и империй, соответственно ($1 \leq n \leq 30\,000$; $0 \leq m \leq 10^5$; $2 \leq d \leq 20$). Гарантируется, что $d \leq n$.

Вторая строка содержит n целых чисел e_i — принадлежности городов к империям ($1 \leq e_i \leq d$).

В следующих m строках содержатся пары чисел v_i и t_i , означающие наличие дороги между городами v_i и t_i .

Формат выходных данных

Выведите d строк: i -я строка должна содержать k_i — минимальное количество штабов, а следом k_i чисел — список номеров городов, в которых эти штабы следует расположить.

Примеры

стандартный ввод	стандартный вывод
4 4 2 1 1 2 2 1 2 1 3 2 4 3 4	1 1 1 4
8 10 3 1 1 1 2 2 3 3 3 5 1 5 2 5 3 6 1 6 2 6 3 6 4 7 1 7 4 8 1	1 3 2 4 5 2 8 6

Задача F. Раздача Фибоначчи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Пожилая команда «Путешественники по разуму» пишет уже свой пятьдесят четвёртый четвертьфинал пенсионерского командного чемпионата мира по программированию.

Программисты Иркутского государственного дома престарелых люди уже не молодые, как и их соперники, поэтому контекст для них длится не стандартные студенческие пять часов, а пять месяцев. Спокойное, размеренное решение задач – залог здоровья.

Соответственно, и количество задач на таком контексте бесконечное. «Путешественники» верны своим принципам, поэтому всегда решают на четвертьфинале количество задач, равное некоторому числу Фибоначчи.

Числа Фибоначчи – это числовая последовательность, задаваемая следующими правилами:

- $F_0 = 0$
- $F_1 = 1$
- $F_i = F_{i-2} + F_{i-1}, i \geq 2$

На четвертьфинале 2020 «Путешественники» решили F_n задач. По правилам чемпионата, они во время тура получили ровно столько же воздушных шариков, по одному за каждую задачу.

Капитан команды, Кинир, решил раздать всем трём участникам команды (включая себя) максимальное возможное равное количество шариков из имеющихся. Однако в конце обнаружилось, что осталось несколько лишних шариков.

А вот сколько именно – предстоит ответить вам.

Формат входных данных

В единственной строке содержится целое число n ($0 \leq n \leq 10^9$), означающее, что команда получила F_n шариков за контекст.

Формат выходных данных

В единственную строку выведите единственное число – ответ на задачу (количество лишних шариков, оставшихся после раздачи всех имеющих поровну в максимально возможном количестве).

Примеры

стандартный ввод	стандартный вывод
2	1
5	2

Замечание

В первом примере команда решила $F_2 = 1$ задач, всем досталось по 0 шариков, 1 остался лишним.

Во втором примере команда решила $F_5 = 5$ задач, всем досталось по 1 шарик, 2 остались лишними, чтобы не создавать неравенства между членами команды.

Задача G. Карты, числа, два заклинания

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В ряд расположены n карт. У каждой карты есть сила, выраженная целым неотрицательным числом. Также существует два заклинания.

Заклинание «Равные шансы» уменьшает в два раза силу карт, у которых эта сила чётная.

Заклинание «Странное событие» уменьшает на 1 силу всех карт, у которых эта сила нечётная.

Обработайте заданную последовательность применения заклинаний, выводя после каждого применения очередного заклинания суммарную силу карт.

Формат входных данных

Первая строка содержит целое число n — размер ряда ($1 \leq n \leq 10^5$).

Вторая строка содержит n целых чисел a_i , разделённых пробелами ($1 \leq a_i \leq 2^{30} - 1$) — начальные силы карт.

Третья строка содержит последовательность применения заклинаний в виде символов «0» (заклинание «Равные шансы») и «1» (заклинание «Странное событие»).

Количество заклинаний от 1 до 10^5 .

Формат выходных данных

После каждого заклинания выведите сумму сил карт в отдельной строке.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5 0110	12 8 8 4
3 1 1 1 01	3 0

Замечания

Чётное число — число, которое при делении на 2 даёт остаток 0.

Нечётное число — число, которое не является чётным.

Задача Н. Гипноз

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Роджер безумно любит кататься на велосипеде. А чтобы велосипед не украли, Роджер пристёгивает его с помощью новомодного гипнотического велосипедного замка.

Гипнотический велосипедный замок можно представить матрицей A размера $n \times n$ (n чётно), содержащей целые числа. Каждый замок разделён на $\frac{n}{2}$ прямоугольных рамок. Более формально, рамочка с номером i представляет собой числовую последовательность B_i , такую, что:

$$B_i = \{A_{i,i}, A_{i,i+1}, \dots, A_{i,n+1-i}, A_{i+1,n+1-i}, \dots, A_{n+1-i,n+1-i}, A_{n+1-i,n-i}, \dots, A_{n+1-i,i}, A_{n-i,i}, \dots, A_{i+1,i}\}.$$

Для введения кода от велосипедного замка каждую рамочку можно вращать по часовой стрелке. Более формально, можно делать циклический сдвиг вправо последовательности, соответствующей нужной рамочке. Из-за таких вращений и происходит гипнотический эффект замка.

Старый замок Роджера то и дело заклинивает, посему он решил приобрести новый. Однако Роджеру хотелось бы и с новым замком иметь возможность вводить его любимые коды. Роджер хотел бы купить новый замок эквивалентный старому.

Два замка эквивалентны, если их рамочки с соответствующими номерами можно привести к абсолютно одинаковому виду одними лишь их вращениями.

Имея два заданных замка — старый и новый — определите, эквивалентны ли они.

Формат входных данных

В первой строке входных данных задано единственное целое чётное число n ($1 \leq n \leq 200$) — размер обоих замков, старого и нового.

Следующие n строк входных данных содержат по n целых чисел, записанных через пробел. j -е число в i -й из этих строк обозначается $O_{i,j}$ и задаёт элемент матрицы, соответствующей старому замку, находящийся в i -й строке и j -м столбце. Причём $1 \leq O_{i,j} \leq 10^9$.

Следующие n строк входных данных содержат по n целых чисел, записанных через пробел. j -е число в i -й из этих строк обозначается $N_{i,j}$ и задаёт элемент матрицы, соответствующей новому замку, находящийся в i -й строке и j -м столбце. Причём $1 \leq N_{i,j} \leq 10^9$.

Формат выходных данных

В единственной строке выходных данных выведите «YES» (без кавычек), если старый и новый замок эквивалентны и «NO» (без кавычек) иначе.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 3 1 2 5 2 4 3 6 1 9 8 7 7 8 9 1 6 4 1 2 5 3 2 3 4 3 2 1	YES
4 1 2 3 4 3 1 2 5 2 4 3 6 1 9 8 7 7 8 9 1 6 1 3 2 5 4 2 3 4 3 2 1	NO

Замечание

В первом примере старый замок имеет две рамочки: $BO_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3\}$ и $BO_2 = \{1, 2, 3, 4\}$. Новый замок имеет две рамочки: $BN_1 = \{7, 8, 9, 1, 2, 3, 1, 2, 3, 4, 5, 6\}$ и $BN_2 = \{4, 1, 2, 3\}$.

BO_1 и BN_1 можно привести к абсолютно одинаковому виду, например, сдвинув BN_1 циклически вправо на 6 позиций. BO_2 и BN_2 можно привести к одинаковому виду сдвинув BO_2 циклически вправо на 1 позицию. Поэтому замки эквивалентны.

Во втором примере внутренние рамочки обоих замков нельзя превратить в одинаковые циклическими сдвигами, поэтому замки не эквивалентны.

Задача I. Круговой марафон

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Антон в очередной раз пригласили принять участие в марафоне, но этот марафон оказался довольно необычным — он проводился на стадионе!

Антон позвали в другой город, поэтому про стадион, на котором он собирается бежать, ему ничего не известно (даже примерно). Чтобы скрасить свое участие в этом довольно скучном событии, он поставил перед собой задачу — узнать длину кругового участка стадиона.

Опыт Антона в забегах позволяет ему с высокой точностью отмерять расстояние и пробежать ровно k метров по треку. Но памятью наш бегун похвастаться не может. Единственное, что он в силах запомнить — это сколько кругов он уже пробежал.

Антон предлагает вам скооперироваться, а именно помочь ему узнать длину беговой дорожки стадиона за **не более чем 100** продвижений по треку.

Протокол взаимодействия

Взаимодействие происходит с помощью запросов.

Вы можете просить Антона пробежать на k метров вперёд с помощью запроса «run k » ($1 \leq k \leq 10^9$). В ответ на такой запрос вы получите единственное число — количество кругов, которое он уже пробежал, включая текущий запрос. Круг считается завершённым, если бегун пробегает стартовую точку или находится в ней.

Гарантируется, что длина участка — положительное целое число, не превышающее 10^9 .

Если вы готовы указать длину кругового участка, то вы можете убедиться в его правильности с помощью запроса «ensure s ». После такого запроса программа должна немедленно завершиться.

При превышении количества запросов (более 100) вы получите вердикт «**Wrong answer**» (при условии завершения работы программы).

Ваше решение может получить вердикт «**Idleness limit exceeded**», если вы ничего не выводите или забываете сбросить буфер вывода.

Для корректного взаимодействия после каждой операции вывода данных вам необходимо выводить **перенос строки**, а также **очищать буфер вывода**, то есть делать следующие операции:

- `cout.flush()` или `fflush(stdout)` в C++;
- `stdout.flush()` в Python;
- `flush(output)` в Pascal;
- см. документацию других языков.

Пример

стандартный ввод	стандартный вывод
1	run 5
1	run 2
2	run 4
3	run 1
	ensure 4

Задача J. Пасьянс по-иркутски

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дед Василий с дальней заимки, коротая долгие зимние вечера, придумал следующий пасьянс: пусть в ряд разложено n карт, на каждой из которых написано по одному числу: a_1, a_2, \dots, a_n . Дед Василий просматривает эти карты слева направо и некоторые из них перекладывает в правый конец ряда. Условием перемещения карты является наличие справа от неё хотя бы одной карты с большим чем у неё значением. Более строго: карта на позиции i и значением a_i будет перемещена вправо, если найдётся позиция $j > i$ такая, что $a_j > a_i$. Пасьянс считается законченным, когда дед Василий доходит до самой правой на данный момент карты.

Иногда пасьянс затягивается уж очень надолго. Поэтому дед Василий задался вопросом: как по исходному расположению карт выяснить, сколько перекладываний придётся сделать.

Формат входных данных

В первой строке находится одно целое число n ($1 \leq n \leq 10^6$) — количество карт в пасьянсе.

Во второй строке содержится n целых чисел a_i через пробел — описание исходного расположения карт на столе слева направо ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Вывести одно число — количество карт, которые придётся переложить прежде чем пасьянс сойдётся.

Пример

стандартный ввод	стандартный вывод
10 3 7 6 8 5 8 2 1 7 6	14

Замечание

- Рассмотрим пример **3 7 6 8 5 8 2 1 7 6**. При просмотре слева направо, можно увидеть, что карты **3 7 6 5 2 1** (в позициях 1 2 3 5 7 8) имеют справа от себя карту с большим значением и поэтому будут перемещены в таком порядке вправо. После перемещения этих карт получится **8 8 7 6 3 7 6 5 2 1** и мы дошли до теперь первой слева карты **6**.
- Продолжим перемещение и получим, что будут перемещены карты **6** и **3** (позиции 4 5 в новом наборе), так как справа от них есть карта со значением **7**, получим **8 8 7 7 6 5 2 1 6 3**.
- Далее будут перемещены карты **5 2 1** (позиции 6 7 8), получим **8 8 7 7 6 6 3 5 2 1**.
- Теперь извлекается **3** с позиции 7 и получится **8 8 7 7 6 6 5 2 1 3**.
- Наконец перемещаются **2 1** (позиции 8 9) и получается **8 8 7 7 6 6 5 3 2 1**.

Если подсчитать количество произведённых перемещений, то получим 14.

Задача К. Shark Attack

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Одним осенним вечером n кальмарам из одномерного мира стала известна печальная новость: к их месту обитания приближается акула!

Акула и кальмары могут двигаться со скоростью 1 метр в секунду. При этом кальмары имеют единое сознание, поэтому в каждый момент времени двигается максимум один кальмар. Если в какой-либо момент времени акула и кальмар находятся в одной и той же позиции (их размерами можно пренебречь), то акула съедает кальмара.

Конечно же, у кальмаров есть убежище, в которое вместится любое количество кальмаров, а акула не пройдет. Таким образом, если кальмар добирается до своего убежища, и над ним не находится акула, то он спасается и ему больше ничего не угрожает.

Вы, как воплощение коллективного сознания кальмаров, хотите спасти как можно больше представителей кальмарного «социума». Каждую секунду Вы можете отдавать приказ движения одному из кальмаров.

Какое максимальное количество кальмаров Вы сможете гарантированно спасти, вне зависимости от действий акулы?

Формат входных данных

В первой строчке находится единственное число n ($1 \leq n \leq 10^5$) — количество кальмаров.

Во второй строчке находится n целых чисел x_1, x_2, \dots, x_n ($|x_i| \leq 10^9$) — координаты кальмаров в одномерном мире в метрах.

В третьей строчке находится единственное целое число y ($|y| \leq 10^9$) — координата акулы в метрах.

В последней строчке находится единственное целое число z ($|z| \leq 10^9$) — координата убежища кальмаров в метрах.

Гарантируется, что ни один кальмар не находится у выхода или с акулой в начале событий.

Формат выходных данных

Выведите единственное число — максимальное количество кальмаров, которое Вы сможете гарантированно спасти.

Примеры

стандартный ввод	стандартный вывод
3 -1 1 3 2 0	1
3 0 0 9 10 1	2

Задача L. Сравнение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Матрица сравнения для двух строк s и t над алфавитом $\{A, C, G, T\}$ определяется следующим образом. Рассмотрим операцию: вставка в строку символа «-». Такой символ можно вставить в начало и конец строки, а также между любой парой имеющихся символов в строке. Вставим произвольное (возможно, нулевое) количество таких символов в s и t так, чтобы их длины стали равны. Затем запишем полученные строки в матрицу друг под другом. Например, для строк АСТ и АГТТ одна из возможных матриц сравнения выглядит так:

$$\begin{array}{cccc} A & C & - & T \\ A & G & T & T \end{array}$$

Вес такой матрицы считается как сумма весов каждого столбца. Вес столбца определяется следующим образом: если символы в столбце совпадают и оба не равны «-», то вес равен $+1$, и -1 в противном случае. Не допускаются матрицы, в котором в одном и том же столбце находятся два символа «-». Например, вес матрицы, приведенной выше, равен $+1 - 1 - 1 + 1 = 0$. Наибольший возможный вес матрицы сравнения для двух строк называется их *схожестью*.

Подстрокой строки $s = s_1, \dots, s_{|s|}$ является строка $s_{ij} = s_i, \dots, s_j$, $1 \leq i \leq j \leq |s|$; подстрока не может быть пустой. Для данных двух строк s и t найдите подстроки s' в строке s и t' в строке t такие, что схожесть s' и t' максимальная среди всех подстрок s и t .

Формат входных данных

В первой строке находятся два разделенных пробелом целых числа n и m ($1 \leq n, m \leq 10^3$) — размеры строк s и t .

Во второй и третьей строках содержатся n и m символов из алфавита $\{A, C, G, T\}$, представляющие строки s и t соответственно. Все символы находятся в верхнем регистре.

Формат выходных данных

Выведите максимальную схожесть строк s' и t' , где s' и t' — подстроки s и t соответственно.

Примеры

стандартный ввод	стандартный вывод
9 7 CACCGTAAA TCCGAAA	5
3 3 ACC TGG	-1

Замечание

Для первого примера входных данных наибольшую схожесть имеют подстроки CCGTAAA и CCGAAA.

Во втором примере любая пара символов из двух строк имеет схожесть -1 , а любые более длинные подстроки имеют меньшую схожесть.