

Задача А. НОПроблемо с восстановлением

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

Даны две последовательности. Найдите их наибольшую общую подпоследовательность (подпоследовательность — это то, что можно получить из данной последовательности вычёркиванием некоторых элементов).

Формат входных данных

В первой строке входного файла записано число N — длина первой последовательности ($1 \leq N \leq 10^3$). Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10^4 по модулю. В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 10^3$). В четвёртой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10^4 по модулю.

Формат выходных данных

В первой строке выведите длину искомой подпоследовательности

Во второй строке выведите через пробел её элементы. Если есть несколько ответов, выведите любой.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 4 2 1 3 5	2 1 3
3 1 2 3 3 1001 1002 1003	0

Задача В. НВП

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Числовая последовательность задана рекуррентной формулой: $a_{i+1} = (k \cdot a_i + b) \bmod m$. Найдите её наибольшую возрастающую подпоследовательность. Если таких последовательностей несколько, можно вывести любую.

Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности n ($1 \leq n \leq 10^5$), начальный элемент последовательности a_1 , параметры k, b, m для вычисления последующих членов последовательности ($1 \leq m \leq 10^4, 0 \leq k < m, 0 \leq b < m, 0 \leq a_1 < m$).

Формат выходных данных

На первой строке выходного файла вы должны вывести количество чисел в найденной вами наибольшей возрастающей подпоследовательности. На следующей строке выведите элементы подпоследовательности, разделяя их пробелами.

Примеры

стандартный ввод	стандартный вывод
5 41 2 1 100	3 41 67 71
7 1 2 1 10	4 1 3 5 7
7 2 2 1 10	3 1 3 5

Задача С. Шахтеры

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

На двух угольных шахтах работают шахтеры. Добыча угля — тяжелая работа, так что шахтерам необходимо доставлять еду прямо в шахты. Каждый раз, когда очередная порция продовольствия доставляется в шахту, шахтеры производят некоторое количество угля. Существует три типа еды для шахтеров: мясо, фрукты и бублики. Шахтеры любят разнообразие в еде и будут работать продуктивнее, если их диета будет разнообразной. Точнее, каждый раз, когда в шахту доставляется новая порция продовольствия, необходимо посмотреть на две предыдущие поставки (или меньше чем две, если их еще не было) и действовать по следующему правилу:

- если вся еда была одинаковой — будет произведена одна тонна угля
- если еды была двух типов — будет произведено две тонны угля
- если трех типов — три тонны угля

Тип и порядок поставок еды известен. Однако, можно выбрать, на какую из двух шахт какую поставку отправить. Поставка не может быть разделена и обязательно должна быть доставлена на первую или вторую шахту. Две шахты не обязательно должны получить одинаковое количество поставок (например, все поставки могут идти на одну шахту). По известному порядку и типу поставок необходимо написать программу, определяющую максимально возможное количество тонн угля, которое можно добыть на обеих шахтах, при наилучшем распределении поставок между первой и второй шахтой.

Формат входных данных

Первая строка содержит число N ($1 \leq N \leq 100\,000$) — количество поставок еды.

Вторая строка содержит N символов — типы поставок в том порядке, в котором они будут осуществляться. Каждый из символов будет большой латинской буквой М (мясо), F (фрукты) или В (бублики).

Формат выходных данных

Выведите единственное число — количество тонн угля, которое можно добыть.

Примеры

стандартный ввод	стандартный вывод
6 MBMFVB	12
16 MMBMBVVVMMMMMBVB	29

Задача D. Расстояние Дамерау-Левенштейна

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

Одной из наиболее распространенных опечаток при наборе текста является перестановка двух соседних символов, например, вместо слова «программа» набрано слово «прогармма». Расстояние Левенштейна не учитывает такие опечатки: при вычислении расстояния Левенштейна одна перестановка будет считаться за два редактирования (например, удаление и вставка символа).

При вычислении расстояния Дамерау-Левенштейна, помимо операций замены, вставки и удаления символа допускается еще операция перестановки двух соседних символов. При этом между переставленными символами нельзя вставлять другие символы.

Определите расстояние Дамерау-Левенштейна для двух данных строк.

Формат входных данных

Программа получает на вход две строки, длина каждой из которых не превосходит 1000 символов, строки состоят только из заглавных латинских букв.

Формат выходных данных

Требуется вывести одно число – расстояние Дамерау-Левенштейна для данных строк.

Пример

стандартный ввод	стандартный вывод
XABCDE ACBYDF	4

Задача Е. Вано в Аграхабе

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

На базаре в Аграхабе проходит акция. Если в одной покупке не менее 10 товаров, то за самый дешёвый из них платить не нужно; если не менее 20 — не нужно платить за два самых дешёвых, и так далее. Более формально, при покупке m товаров не оплачиваются $\lfloor \frac{m}{10} \rfloor$ самых дешёвых из них, а остальные оплачиваются по их цене. Если $m < 10$, то скидки нет и все товары оплачиваются полностью.

Вано пришёл в лавку и выложил перед торговцем в ряд n товаров. Товары пронумерованы от 1 до n в том порядке, в котором они лежат; цена i -го товара равна a_i . Менять порядок товаров Вано не хочет, но может разбить ряд на несколько покупок, каждая из которых состоит из подряд идущих товаров. Каждый товар должен попасть ровно в одну покупку, а каждая покупка оплачивается отдельно с учётом акции.

Помогите Вано узнать, какую минимальную суммарную стоимость могут иметь все его покупки.

Формат входных данных

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

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — цены товаров в порядке их расположения в ряду.

Формат выходных данных

Выведите одно целое число — минимальную суммарную стоимость всех покупок.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	6
12 1 1 10 10 10 10 10 9 10 10 10	92

Замечание

В первом примере товаров меньше десяти, скидки нет, и Вано платит $1 + 2 + 3 = 6$.

Во втором примере выгодно купить первые два товара по отдельности, а остальные товары — одной покупкой. В ней самый дешёвый товар стоит 9 и достаётся бесплатно, так что всего Вано платит 92.

Задача F. Футбол

Имя входного файла: `football.in`
Имя выходного файла: `football.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Футбольный клуб «Инжир» не имеет проблем с финансированием, поэтому приобрел N полевых игроков. После долгих месяцев тренировок и тестов тренерский коллектив подробно изучил каждого игрока и собирается выбрать оптимальный стартовый состав.

Каждый игрок характеризуется тремя целыми неотрицательными числами: d_i — умение играть в защите, m_i — умение играть в полузащите и a_i — умение играть в нападении. Играть было решено по схеме 5-3-2. Это означает, что в основном составе будет 5 игроков защиты, 3 полузащитника и 2 нападающих. Цель тренерского штаба — выбрать 10 игроков так, чтобы сумма соответствующих умений была максимальна (для игрока обороны значение имеет только его умение играть в защите, аналогично для других позиций). Разумеется, каждый игрок может играть не более чем на одной позиции. Ваша задача — помочь тренерскому штабу подобрать оптимальный состав.

Формат входных данных

В первой строке входного файла находится одно целое число N ($10 \leq N \leq 5 \cdot 10^5$) — количество игроков в клубе. Следующие N строк файла содержат по три целых числа каждая d_i , m_i и a_i ($0 \leq d_i, m_i, a_i \leq 10^7$) — умения футболиста с номером i играть в обороне, полузащите и нападении соответственно.

Формат выходных данных

Выходной файл должен содержать ровно три строки. Первая строка должна содержать пять чисел — номера игроков обороны. Вторая строка должна содержать ровно три числа — номера игроков полузащиты. Третья строка должна содержать ровно два числа — номера игроков нападения. Если оптимальных ответов несколько, можно вывести любой.

Примеры

football.in	football.out
10 1 2 3 2 3 4 3 4 5 4 5 6 5 6 7 6 7 8 7 8 9 8 9 10 9 10 1 10 1 2	10 9 8 7 6 3 4 5 1 2
10 1 5 1 3 2 2 3 3 5 2 1 3 4 4 5 2 0 2 1 1 1 4 5 1 1 3 3 4 5 4	10 5 2 6 7 1 8 9 3 4

Задача G. Общие подпоследовательности

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

Вам даны две целочисленные последовательности S и T длины N и M соответственно, состоящие из целых чисел от 1 до 10^5 (включительно). Сколько есть пара подпоследовательностей S и T таких, что эти две подпоследовательности совпадают?

Здесь подпоследовательность A - это последовательность, полученная путем удаления нуля или более элементов из A и объединения оставшихся элементов без изменения порядка.

Как для S , так и для T мы считаем две подпоследовательности различными, если наборы индексов удаленных элементов различны, даже если подпоследовательности совпадают как наборы чисел. Поскольку ответ может быть огромным, выведите число по модулю $10^9 + 7$.

Формат входных данных

В первой строке даны два числа — длины N и M ($N, M \leq 2000$). Во второй строке через пробел задаются числа последовательности S , в третьей — последовательности T .

Формат выходных данных

Выведите число пар подпоследовательностей S и T таких, чтобы они совпадали. Ответ выведите по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 2 1 3 3 1	3
2 2 1 1 1 1	6
4 4 3 4 5 6 3 4 5 6	16
10 9 9 6 5 7 5 9 8 5 6 7 8 6 8 5 5 7 9 9 7	191

Задача Н. Турнир Вано

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вано работает на регистрации ежегодного студенческого турнира по настольному теннису. Участники турнира — N студентов — стоят в одной очереди. Команды формируются по порядку очереди: первые K студентов образуют первую команду, следующие K — вторую, и так далее.

Вано знает место каждого студента в рейтинге: студент, стоящий в очереди i -м, занимает место v_i . Чем меньше место, тем сильнее студент. Все места различны.

Вано хочет, чтобы в первой команде оказались K сильнейших студентов, то есть K студентов с наименьшими местами в рейтинге, во второй команде — следующие K по силе студентов, и так далее. Порядок студентов внутри одной команды Вано не важен.

Пока его напарник на перерыве, Вано за одну минуту может попросить одного студента выйти из очереди и встать либо в её начало, либо сразу за любым другим студентом. Помогите Вано выяснить, за какое минимальное число минут он сможет собрать команды нужным образом.

Формат входных данных

Первая строка содержит два целых числа N и K ($1 \leq K \leq N \leq 5000$) — количество студентов и размер команды. Гарантируется, что N делится на K .

Вторая строка содержит N попарно различных целых чисел $v_1, v_2, \dots, v_N$ ($1 \leq v_i \leq 10^9$) — места студентов в рейтинге в порядке их следования в очереди.

Формат выходных данных

Выведите одно целое число — минимальное количество минут.

Примеры

стандартный ввод	стандартный вывод
4 1 9 12 5 13	1
6 2 16 2 1 7 5 10	1
6 3 7 9 8 3 6 5	3

Замечание

В первом примере достаточно поставить в начало очереди студента с местом 5. Во втором достаточно поставить студента с местом 16 сразу за студентом с местом 10. В третьем Вано по очереди переставляет в начало очереди студентов с местами 5, 6 и 3, и очередь принимает вид 3, 6, 5, 7, 9, 8.

Задача I. Друг Ваню заикается

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	32 мегабайта

Друг Ваню «заболел»: он сильно заикается, и единственные слова, которые он произносит, — числа. Ваню заметил странную повторяемость в заиканиях друга и подозревает, что тот просто притворяется, чтобы не ходить в школу.

Напомним, что *подпоследовательность* получается из последовательности вычёркиванием любых её элементов (возможно, ни одного) без изменения порядка остальных. Например, 1, 1, 7, 5 является подпоследовательностью последовательности 1, 3, 1, 7, 6, 6, 5, 5.

Назовём последовательность *заиканием*, если она имеет вид $x_1, x_1, x_2, x_2, \dots, x_k, x_k$ при некотором целом неотрицательном k , причём значения x_i могут совпадать между собой. Иными словами, в заикании каждое число произнесено два раза подряд. Например, 1, 1, 1, 1, 3, 3 является заиканием, а 1, 1, 7, 5 — нет. При этом два одинаковых числа, образующих пару в заикании, в исходной последовательности не обязаны стоять рядом: так, 1, 1, 1, 1, 3, 3 является подпоследовательностью последовательности 1, 2, 1, 2, 1, 2, 1, 3, 3.

Друг Ваню произнёс два высказывания: последовательность $a_1, \dots, a_n$ и последовательность $b_1, \dots, b_m$. Чтобы разоблачить друга, Ваню хочет найти самое длинное заикание, которое встречается в обоих высказываниях, то есть является подпоследовательностью и первой, и второй последовательности. Помогите Ваню найти длину такого заикания.

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leq n, m \leq 15\,000$) — длины первого и второго высказываний друга Ваню.

Вторая строка содержит n целых чисел $a_1, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — первое высказывание.

Третья строка содержит m целых чисел $b_1, \dots, b_m$ ($1 \leq b_i \leq 10^9$) — второе высказывание.

Формат выходных данных

Выведите одно неотрицательное целое число — наибольшую длину заикания, которое является подпоследовательностью и первой, и второй последовательности.

Примеры

стандартный ввод	стандартный вывод
7 9 1 2 2 3 1 1 1 2 4 2 3 1 2 4 1 1	4
5 4 42 42 42 42 42 42 42 42 42	4

Замечание

В первом примере самое длинное общее заикание — 2, 2, 1, 1.

Во втором примере в первой последовательности пять одинаковых чисел, а во второй — четыре, поэтому общим заиканием длины 4 является 42, 42, 42, 42.

Задача J. Массив Вано

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вано вместе с двумя друзьями, Нино и Леваном, участвует в командной олимпиаде по программированию. Задачи тура выписаны в ряд в порядке номеров, и этот ряд Вано называет своим массивом. Чтобы не терять время, ребята заранее оценили задачи: каждый из троих выставил каждой задаче целое число от 1 до 5. Оценка 1 означает «хе-хе-хе», 2 — «давай сюда!», 3 — «ну ладно», 4 — «хмммм...», а 5 — «ты с ума сошёл?».

Теперь массив нужно разрезать на три непустых куска из подряд идущих задач, так чтобы каждая задача попала ровно в один кусок. Каждый из ребят получает ровно один кусок и решает все задачи в нём. Кому какой кусок достанется — ребята выбирают произвольно: левым, средним и правым куском могут оказаться кто угодно из троих.

Стоимостью разрезания назовём сумму оценок по всем задачам, где для каждой задачи берётся оценка того, кому эта задача досталась. Иными словами, оценки двух других ребят для этой задачи в сумму не входят.

Помогите Вано разрезать массив и раздать куски так, чтобы стоимость оказалась как можно меньше, и найдите эту стоимость.

Формат входных данных

В первой строке записано целое число N ($3 \leq N \leq 150\,000$) — количество задач.

В каждой из следующих трёх строк записано по N целых чисел $a_{j,1}, a_{j,2}, \dots, a_{j,N}$ ($1 \leq a_{j,i} \leq 5$) — оценки сложности задач $1, 2, \dots, N$, выставленные j -м из ребят. Первая из этих строк содержит оценки Вано, вторая — оценки Нино, третья — оценки Левана.

Формат выходных данных

Выведите одно число — минимально возможную стоимость.

Примеры

стандартный ввод	стандартный вывод
3 1 3 3 1 1 1 1 2 3	4
7 3 3 4 1 3 4 4 4 2 5 1 5 5 4 5 5 1 3 4 4 4	19

Замечание

В первом примере первый кусок состоит из задачи 1 и достаётся Вано, второй кусок состоит из задачи 2 и достаётся Левану, третий кусок состоит из задачи 3 и достаётся Нино. Стоимость равна $1 + 2 + 1 = 4$.

Во втором примере первые две задачи можно отдать Нино, третью — Левану, а оставшиеся четыре — Вано. Стоимость равна $(4 + 2) + 1 + (1 + 3 + 4 + 4) = 19$.

Задача К. Башня Ваню

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

У Ваню есть кубики одинакового размера. На каждом кубике написано положительное целое число. Ваню поставил из всех кубиков одну высокую башню. Кубики в башне пронумерованы снизу вверх целыми числами от 1 до n , на i -м кубике написано число a_i .

Цель игры — чтобы как можно больше кубиков оказались на своих местах. Кубик с числом v стоит на своём месте, если он находится на высоте v . Самый нижний кубик башни находится на высоте 1, следующий за ним — на высоте 2, и так далее. Иными словами, кубик стоит на своём месте, если число на нём равно количеству кубиков башни, лежащих не выше него, считая сам этот кубик.

Ваню решил осторожно вынуть из башни некоторые кубики, стараясь, чтобы башня не упала. Можно вынуть любой набор кубиков. После этого оставшиеся кубики опускаются вниз, сохраняя прежний порядок, так что в башне не остаётся пропусков.

Помогите Ваню выбрать, какие кубики убрать из башни, чтобы на своих местах оказалось как можно больше кубиков.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 100\,000$) — количество кубиков в башне.

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^6$) — числа на кубиках, записанные от нижнего кубика к верхнему. Числа на разных кубиках могут совпадать.

Формат выходных данных

В первой строке выведите одно целое число k — количество кубиков, которые нужно убрать. Во второй строке выведите k различных целых чисел от 1 до n — номера убираемых кубиков в исходной башне, в любом порядке. Если $k = 0$, вторая строка должна быть пустой.

Количество убираемых кубиков не обязано быть наименьшим возможным: важно лишь, сколько кубиков окажется на своих местах. Если ответов несколько, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
5	1
1 1 2 5 4	1

Замечание

В примере в башне пять кубиков. Если убрать кубик с номером 1, останутся кубики с числами 1, 2, 5, 4, которые окажутся на высотах 1, 2, 3, 4 соответственно. На своих местах будут стоять кубики с числами 1, 2 и 4, то есть три кубика. Больше трёх кубиков на своих местах оказаться не может.

Задача L. Экспериментатор

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

На старости лет один профессор загорелся идеей исследования на прочность транзисторов «Кд521(2)», К сожалению, ему не удалось привлечь на помощь никого из коллег, поэтому проводить измерения придется самостоятельно. Но это не пугает профессора. В шкафу профессор обнаружил t транзисторов данной модели, оставшихся со старых времен, и решил использовать их для экспериментов.

После некоторых размышлений был выбран следующий способ проведения измерений: профессор собирается, перемещаясь по пожарной лестнице, сбрасывать транзисторы с различных этажей. Таким образом он планирует определить, при падении с какого минимального этажа транзистор разбивается. При этом профессор уверен, что транзистор не может выдержать падение с последнего этажа, однако падение с высоты человеческого роста (то есть когда профессор находится на первом этаже) не причиняет транзистору вреда. Известно, что все транзисторы абсолютно одинаковые, и если транзистор разбивается при падении с некоторого этажа, то он разбивается и при падении со всех этажей с большим номером. Разбившиеся транзисторы снова использовать нельзя, а если транзистор остался целым после падения, его можно использовать повторно. Для того, чтобы поднять оставшийся целым транзистор, профессору надо спуститься на первый этаж. Оказавшись на первом этаже, профессор может поднять все лежащие там транзисторы.

Годы профессора уже дают о себе знать, поэтому он хочет минимизировать суммарное расстояние, которое ему придется подниматься по лестнице. Но, возраст дает и определенные преимущества - сняв очки, профессор может с любого этажа определить, разбился транзистор или нет. Изначально профессор находится на первом этаже, и у него имеется m транзисторов. В доме, в котором живет профессор, n этажей. Найдите минимальное число этажей, которое профессору в худшем случае придется подниматься вверх по лестнице во время проведения экспериментов.

Формат входных данных

Во входном файле заданы два целых числа - высота дома n ($2 \leq n \leq 50$) и количество транзисторов m ($1 \leq m \leq 10$).

Формат выходных данных

В выходной файл выведите единственное число - минимальное расстояние в этажах, которое в худшем случае придется подниматься вверх по лестнице профессору во время эксперимента.

Примеры

стандартный ввод	стандартный вывод
2 1	0
3 1	1
18 1	136
17 3	23