

Задача А. Пересечение множеств

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Даны два списка чисел, которые могут содержать до 10000 чисел каждый. Выведите все числа, которые входят как в первый, так и во второй список в порядке возрастания.

Формат входных данных

Вводятся два списка целых чисел. Все числа каждого списка находятся на отдельной строке.

Формат выходных данных

Выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1 3 2 4 3 2	2 3

Задача В. Коммерческий калькулятор

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Фирма OISAC выпустила новую версию калькулятора. Этот калькулятор берет с пользователя деньги за совершаемые арифметические операции. Стоимость каждой операции в долларах равна 5% от числа, которое является результатом операции.

На этом калькуляторе требуется вычислить сумму N натуральных чисел (числа известны). Нетрудно заметить, что от того, в каком порядке мы будем складывать эти числа, иногда зависит, в какую сумму денег нам обойдется вычисление суммы чисел (тем самым, оказывается нарушен классический принцип «от перестановки мест слагаемых сумма не меняется»).

Например, пусть нам нужно сложить числа 10, 11, 12 и 13. Тогда если мы сначала сложим 10 и 11 (это обойдется нам в \$1.05), потом результат — с 12 (\$1.65), и затем — с 13 (\$2.3), то всего мы заплатим \$5, если же сначала отдельно сложить 10 и 11 (\$1.05), потом — 12 и 13 (\$1.25) и, наконец, сложить между собой два полученных числа (\$2.3), то в итоге мы заплатим лишь \$4.6.

Напишите программу, которая будет определять, за какую минимальную сумму денег можно найти сумму данных N чисел.

Формат входных данных

Во входном файле записано число N ($2 \leq N \leq 100000$). Далее идет N натуральных чисел, которые нужно сложить, каждое из них не превышает 10000.

Формат выходных данных

В выходной файл выведите, сколько денег нам потребуется на нахождение суммы этих N чисел. Результат должен быть выведен с двумя знаками после десятичной точки.

Примеры

стандартный ввод	стандартный вывод
4 10 11 12 13	4.60
2 1 1	0.10

Задача С. Игра в пьяницу

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт – проигрывает.

Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза").

Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды).

Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.

Формат входных данных

Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами — номера карт первого игрока, вторая — аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

Формат выходных данных

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово `first` или `second`, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 10^6 ходов игра не заканчивается, программа должна вывести слово `botva`.

Примеры

стандартный ввод	стандартный вывод
1 3 5 7 9 2 4 6 8 0	second 5

Задача D. Списки по классам

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Формат входных данных

В каждой строке сначала записан номер класса (число, равное 9, 10 или 11), затем (через пробел) – фамилия ученика. Общее число строк в файле не превосходит 100000. Длина каждой фамилии не превосходит 50 символов.

Формат выходных данных

Необходимо вывести список школьников по классам: сначала всех учеников 9 класса, затем – 10, затем – 11. Внутри одного класса порядок вывода фамилий должен быть таким же, как на входе.

Примеры

стандартный ввод	стандартный вывод
9 Иванов	9 Иванов
10 Петров	9 Григорьев
11 Сидоров	9 Сергеев
9 Григорьев	10 Петров
9 Сергеев	10 Яковлев
10 Яковлев	11 Сидоров

Задача Е. Хеширование с удалением

Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных типа “множество строк”. Хранимые строки – непустые последовательности длиной не более 10 символов, состоящие из строчных латинских букв. Структура данных должна поддерживать операции добавления строки в множество, удаления строки из множества и проверки принадлежности данной строки множеству. Максимальное количество элементов в хранимом множестве не превосходит 10^6 .

Формат входных данных

Каждая строка входных данных задает одну операцию над множеством. Запись операции состоит из типа операции и следующей за ним через пробел строки, над которой проводится операция. Тип операции – один из трех символов: + означает добавление данной строки в множество; - означает удаление строки из множества; ? означает проверку принадлежности данной строки множеству. Общее количество операций во входном файле не превосходит 10^6 . Список операций завершается строкой, в которой записан один символ # – признак конца входных данных. При добавлении элемента в множество НЕ ГАРАНТИРУЕТСЯ, что он отсутствует в этом множестве. При удалении элемента из множества НЕ ГАРАНТИРУЕТСЯ, что он присутствует в этом множестве.

Формат выходных данных

Программа должна вывести для каждой операции типа ? одну из двух строк YES или NO, в зависимости от того, встречается ли данное слово в нашем множестве.

Примеры

стандартный ввод	стандартный вывод
+ hello	YES
+ bye	NO
? bye	YES
- bye	
? bye	
? hello	
#	

Задача F. Экспресс 20/19

Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Система, связывающая столицу Флатландии с соседними городами с помощью электричек, сильно устарела. Для её модернизации было принято решение пустить из столицы до одной из станций электричку-экспресс.

Всего в железнодорожной сети Флатландии n станций, пронумерованных целыми числами от 1 до n , сама столица имеет номер 1. Всего существует m односторонних перегонов между станциями. По каждому из них экспресс может проехать от станции с меньшим номером до станции с большим номером. Для каждого перегона известно время, за которое экспресс проезжает этот перегон.

Маршрутом будем называть некоторую последовательность перегонов, такую, что первый перегон начинается в столице, и для любых двух последовательных перегонов начало второго совпадает с концом первого. Время проезда по маршруту равно суммарному времени прохождения всех этих перегонов, временем стоянки на станциях следует пренебречь.

Министерство транспорта Флатландии планирует рассмотреть несколько вариантов маршрута экспресса от столицы. Каждый вариант характеризуется номером станции, до которой следует пустить экспресс, и ориентировочным временем проезда по маршруту. При этом, в министерстве понимают, что точно выполнить требование о времени проезда может оказаться невозможно. Поэтому они используют для оценки допустимости маршрута величину p : маршрут с временем проезда x является допустимым для ориентировочного времени r , если $r \leq x \leq \frac{p}{p-1} \cdot r$.

Требуется написать программу, которая по описанию железнодорожной сети Флатландии и вариантам маршрута определяет для каждого варианта, существует ли допустимый для этого варианта маршрут экспресса.

Формат входных данных

В этой задаче один запуск программы должен обрабатывать несколько тестов.

В первой строке ввода находится единственное целое число t — количество тестов ($1 \leq t \leq 1000$). В следующих строках следует описание этих тестов друг за другом в следующем формате.

В первой строке находятся четыре целых числа n, m, q, p — количество станций, количество перегонов, количество вариантов, которые требуется рассмотреть, и параметр, задающий допустимое превышение ориентировочного времени проезда маршрута ($2 \leq n \leq 500000$, $1 \leq m \leq 500000$, $1 \leq q \leq 500000$, $2 \leq p \leq 20$).

В следующих m строках находятся по три целых числа, описывающих i -й перегон: v_i, u_i, d_i — станция отправления, станция прибытия и время, за которое экспресс проезжает этот перегон ($1 \leq v_i < u_i \leq n$, $1 \leq d_i \leq 10^{11}$). Гарантируется, что для любой станции существует хотя бы один маршрут из столицы, заканчивающийся в ней.

В следующих q строках находятся по два целых числа f_i и r_i — требование проверить существование допустимого маршрута до станции f_i с ориентировочным временем проезда r_i ($2 \leq f_i \leq n$, $1 \leq r_i \leq 10^{17}$).

Гарантируется, что каждая из сумм значений n , значений m и значений q по всем тестам одних входных данных не превосходит 500000.

Формат выходных данных

Выведите t строк, по одной для каждого заданного во входных данных теста.

В качестве ответа на тест выведите строку s длины q , состоящую из символов 0 и 1. Символ s_i должен быть равен 1, если для i -го варианта существует допустимый маршрут, то есть маршрут до станции f_i , время проезда по которому лежит в отрезке $[r_i, \frac{p}{p-1} \cdot r_i]$, и 0 — в противном случае.

Примеры

стандартный ввод	стандартный вывод
2 3 3 5 20 1 2 20 2 3 1 1 3 10 2 19 2 20 3 20 3 21 3 9 7 10 5 5 1 2 15 1 3 10 2 4 21 3 4 30 2 5 14 3 5 31 4 6 3 5 6 14 1 7 39 5 7 13 7 42 7 43 7 44 5 39 6 44	11110 10111
1 4 6 5 2 1 2 1 2 3 1 3 4 1 1 2 70 2 3 120 3 4 4 4 90 4 2 4 10 4 37 2 34	11010

Задача G. Уроки физкультуры

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Сегодня в 5-А классе праздник — урок физкультуры. Традиционно ребята в это время после небольшой разминки играют в футбол. Коля очень любит футбол, но сегодня, проходя мимо учительской, он случайно услышал, что несколько самых сильных школьников из 5-А во время урока физкультуры должны помочь разгружать новые парты. Что же делать? Ведь Коля предпочитает поиграть в футбол!

В голове Коли моментально созрел план. Коля знает, что в 5-А классе N школьников. Он хочет воспользоваться тем, что учитель физкультуры Иван Петрович на уроках вместо обычной расстановки школьников в шеренгу по росту практикует расстановку «по силе». Для этого Иван Петрович сначала расставляет N школьников по росту, а затем $(N - 1)$ раз проходит вдоль шеренги слева направо, каждый раз начиная с самого левого (первого) школьника и заканчивая предпоследним школьником справа. Проходя мимо школьника, который стоит на i -м месте ($1 \leq i \leq N - 1$), Иван Петрович просит его помериться силой с соседом справа, который стоит на $(i + 1)$ -м месте. Если школьник, стоящий левее, оказывается сильнее своего соседа справа, то они меняются местами. Если же силы школьников оказываются равны, либо слева стоит более слабый школьник, то школьники остаются на своих местах. После этого Иван Петрович просит помериться силой школьников, стоящих на местах $(i + 1)$ и $(i + 2)$, и т. д., заканчивая каждый проход школьниками, которые стоят на местах $(N - 1)$ и N . При любом сравнении «по силе» все школьники, включая Колю, показывают свою реальную силу, так как не хотят прослыть слабаками.

Для увеличения шансов поиграть в футбол, Коля хотел бы оказаться как можно левее в получившейся шеренге. Силу каждого из своих одноклассников он знает. По предыдущим занятиям Коля заметил, что если присесть «завязывать шнурки» при каком-либо из проходов учителя, то Иван Петрович во время такого прохода не будет просить Колю мериться силой ни с соседом слева, ни с соседом справа, и, тем более, не будет просить мериться силой соседей Коли, так как они не стоят рядом. Однако, чтобы сохранить силы для игры в футбол, Коля не может присесть более k раз.

Требуется написать программу, которая определит, как нужно действовать Коле, чтобы после проведения расстановки «по силе» оказаться в шеренге как можно левее.

Формат входных данных

В первой строке входных данных задаются три целых числа: N — число школьников в классе, p — место Коли в шеренге по росту, и k — количество приседаний, которое Коля может сделать, не потеряв при этом способность играть в футбол ($2 \leq N \leq 100000$, $1 \leq p \leq N$, $1 \leq k \leq N - 1$).

Во второй строке задаются целые числа a_1, a_2, \dots, a_N ($1 \leq a_i \leq 10^9$). Число a_i показывает силу школьника, который стоит на i -м месте по росту. Школьник, который стоит на i -м месте в расстановке по росту, сильнее школьника, который стоит j -м месте в расстановке по росту, если $a_i > a_j$.

Формат выходных данных

В первой строке выведите самую левую позицию в шеренге, в которой может оказаться Коля после построения школьников «по силе». Во второй строке выведите любую из возможных стратегий Коли, приводящую к этому результату. Стратегия выводится в виде строки из $(N - 1)$ символов, j -й символ этой строки должен быть равен символу «+», если на j -м проходе Коле необходимо присесть, и символу «-» — в противном случае.

Примеры

стандартный ввод	стандартный вывод
10 5 6 10 1 8 3 6 5 4 7 2 9	5 --+++++

Задача Н. Поможем дикой природе

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Фонд изучения дикой природы в течение t лет ежегодно выделяет денежные гранты в поддержку исследований северной фауны. На гранты претендуют три организации, одна из которых занимается изучением тюленей, вторая — оленей, третья — белых медведей.

Для упрощения бухгалтерского учёта фонд принял следующие решения:

- размер любого гранта в денежных единицах должен быть степенью числа 2, то есть равен 2^k для некоторого целого $k \geq 0$;
- все гранты, получаемые одной организацией в одном году, должны иметь различные размеры.

В i -м году фонд планирует полностью распределить n_i денежных единиц, выделенных на гранты. Сравнивать результативность использования средств возможно только для грантов одинакового размера, выделенных каждой из трёх организаций. Такие гранты называются целевыми. Распределение денежных единиц на гранты между тремя организациями считается оптимальным, если как можно большая часть общей суммы выделена на целевые гранты.

Например, если в текущем году на все гранты выделено 47 денежных единиц, то оптимальным вариантом распределения будет: выделить каждой из организаций целевые гранты размерами по 2 и 8 денежных единиц, что составит в сумме 30 единиц. Остальные 17 единиц можно распределить, например, выделив первой организации 16 денежных единиц, а третьей — 1 денежную единицу. Выделить более 30 денежных единиц на целевые гранты, распределяя 47 денежных единиц, нельзя.

Требуется написать программу, которая по заданной в i -м году общей сумме грантов n_i определяет, сколько денежных единиц следует выделить каждой из трёх организаций при оптимальном распределении грантов.

Более формально, требуется найти такие три числа a , b и c , что $a + b + c = n$, при этом требуется максимизировать $a \& b \& c$, где $\&$ обозначает побитовое «И».

Формат входных данных

В первой строке входных данных записано целое число t — количество лет ($1 \leq t \leq 100$). В каждой из последующих t строк записано целое число n_i — общая сумма грантов, которую необходимо полностью распределить в i -м году.

Формат выходных данных

Выходные данные должны содержать t строк по три целых числа в каждой — суммы грантов, которые следует выделить каждой из трёх организаций в соответствующий год. Если оптимальных вариантов распределения несколько, необходимо вывести любой из них.

Задача I. Правильная скобочная последовательность

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Рассмотрим последовательность, состоящую из круглых, квадратных и фигурных скобок. Программа должна определить, является ли данная скобочная последовательность правильной.

Пустая последовательность является правильной. Если A – правильная, то последовательности (A) , $[A]$, $\{A\}$ – правильные. Если A и B – правильные последовательности, то последовательность AB – правильная.

Формат входных данных

В единственной строке записана скобочная последовательность, содержащая не более 100000 скобок.

Формат выходных данных

Если данная последовательность правильная, то программа должна вывести строку `yes`, иначе строку `no`.

Примеры

стандартный ввод	стандартный вывод
<code>() []</code>	<code>yes</code>

Задача J. Постфиксная запись

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как $A B +$. Запись $B C + D *$ обозначает привычное нам $(B + C) * D$, а запись $A B C + D * +$ означает $A + (B + C) * D$. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Формат входных данных

В единственной строке записано выражение в постфиксной записи, содержащее цифры и операции $+$, $-$, $*$. Числа и операции разделяются пробелами. В конце строки может быть произвольное количество пробелов.

Формат выходных данных

Необходимо вывести значение записанного выражения.

Примеры

стандартный ввод	стандартный вывод
8 9 + 1 7 - *	-102

Задача К. Гоблины и шаманы

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Гоблины Мглистых гор очень любях ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

Формат входных данных

В первой строке входных данных записано число N ($1 \leq N \leq 10^5$) - количество запросов к программе. Следующие N строк содержат описание запросов в формате:

- "+ i " - гоблин с номером i ($1 \leq i \leq N$) встает в конец очереди.
- "* i " - привилегированный гоблин с номером i встает в середину очереди.
- "-" - первый гоблин из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

Формат выходных данных

Для каждого запроса типа "-" программа должна вывести номер гоблина, который должен уйти к шаманам.

Примеры

стандартный ввод	стандартный вывод
7	1
+ 1	2
+ 2	3
-	
+ 3	
+ 4	
-	
-	

Задача L. Контейнеры

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

На складе хранятся контейнеры с товарами N различных видов. Все контейнеры составлены в N стопок. В каждой стопке могут находиться контейнеры с товарами любых видов (стопка может быть изначально пустой).

Автопогрузчик может взять верхний контейнер из любой стопки и поставить его сверху в любую стопку. Необходимо расставить все контейнеры с товаром первого вида в первую стопку, второго вида – во вторую стопку и т.д.

Программа должна вывести последовательность действий автопогрузчика или сообщение о том, что задача решения не имеет.

Формат входных данных

В первой строке входных данных записано одно натуральное число N , не превосходящее 500. В следующих N строках описаны стопки контейнеров: сначала записано число k_i – количество контейнеров в стопке, а затем k_i чисел – виды товара в контейнерах в данной стопке, снизу вверх. В каждой стопке вначале не более 500 контейнеров (в процессе переноса контейнеров это ограничение может быть нарушено).

Формат выходных данных

Программа должна вывести описание действий автопогрузчика: для каждого действия напечатать два числа – из какой стопки брать контейнер и в какую стопку класть. (**Обратите внимание, что минимизировать количество операций автопогрузчика не требуется.**) Если задача не имеет решения, необходимо вывести одно число 0. Если контейнеры изначально правильно размещены по стопкам, то выводить ничего не нужно.

Примеры

стандартный ввод	стандартный вывод
3	1 2
4 1 2 3 2	1 3
0	1 2
0	

Задача М. Контрольная по ударениям

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Учительница задала Пете домашнее задание — в заданном тексте расставить ударения в словах, после чего поручила Васе проверить это домашнее задание. Вася очень плохо знаком с данной темой, поэтому он нашел словарь, в котором указано, как ставятся ударения в словах. К сожалению, в этом словаре присутствуют не все слова. Вася решил, что в словах, которых нет в словаре, он будет считать, что Петя поставил ударения правильно, если в этом слове Петей поставлено ровно одно ударение.

Оказалось, что в некоторых словах ударение может быть поставлено больше, чем одним способом. Вася решил, что в этом случае если то, как Петя поставил ударение, соответствует одному из приведенных в словаре вариантов, он будет засчитывать это как правильную расстановку ударения, а если не соответствует, то как ошибку.

Вам дан словарь, которым пользовался Вася и домашнее задание, сданное Петей. Ваша задача — определить количество ошибок, которое в этом задании насчитает Вася.

Формат входных данных

Вводится сначала число N — количество слов в словаре ($0 \leq N \leq 20000$).

Далее идет N строк со словами из словаря. Каждое слово состоит не более чем из 30 символов. Все слова состоят из маленьких и заглавных латинских букв. В каждом слове заглавная ровно одна буква — та, на которую попадает ударение. Слова в словаре расположены в алфавитном порядке. Если есть несколько возможностей расстановки ударения в одном и том же слове, то эти варианты в словаре идут в произвольном порядке.

Далее идет упражнение, выполненное Петей. Упражнение представляет собой строку текста, суммарным объемом не более 300000 символов. Строка состоит из слов, которые разделяются между собой ровно одним пробелом. Длина каждого слова не превышает 30 символов. Все слова состоят из маленьких и заглавных латинских букв (заглавными обозначены те буквы, над которыми Петя поставил ударение). Петя мог по ошибке в каком-то слове поставить более одного ударения или не поставить ударения вовсе.

Формат выходных данных

Выведите количество ошибок в Петинем тексте, которые найдет Вася.

Примеры

стандартный ввод	стандартный вывод
4 cAnnot cannOt fOund pAge thE pAge cAnnot be fouNd	2
4 cAnnot cannOt fOund pAge The PAGE cannot be found	4