Задача A. Права доступа

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

В файловую систему одного суперкомпьютера проник вирус, который сломал контроль за правами доступа к файлам. Для каждого файла N_i известно, с какими действиями можно к нему обращаться:

- запись (`W`),
- чтение (`R`),
- запуск (`X`).

Вам требуется восстановить контроль над правами доступа к файлам (ваша программа для каждого запроса должна будет возвращать «`OK`» если над файлом выполняется допустимая операция, или же «`Access denied`», если операция недопустима).

Формат входных данных

В первой строке входного файла содержится число N ($1 \leqslant N \leqslant 10\,000$) — количество файлов, содержащихся в данной файловой системе.

В следующих N строчках содержатся имена файлов, состоящие из маленьких латинских букв, цифр, точек и символов подчёркивания, и допустимых с ними операций, разделенные пробелами. Длина имени файла не превышает 15 символов.

Далее указано число M ($1 \leqslant M \leqslant 50\,000$) — количество запросов к файлам.

В последних M строках указан запрос вида «Операция Файл». К одному и тому же файлу может быть применено любое количество запросов.

Формат выходных данных

Для каждого из M запросов нужно вывести в отдельной строке «`Access denied`» или «`OK`».

Примеры

access.in	access.out
4 helloworld.exe R X pinglog W R nya R goodluck X W R 5 read nya write helloworld.exe execute nya read pinglog write pinglog	OK Access denied Access denied OK OK

Задача B. Минимум и максимум

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

Пусть есть множество целых чисел. Необходимо реализовать структуру данных для их хранения, поддерживающую следующие операции: `GetMin` — извлечение минимума, `GetMax` — извлечение максимума, `Insert(N)` — добавление числа в множество.

Формат входных данных

В первой строке входного файла записано одно целое число N ($1 \leqslant N \leqslant 100\,000$) — число запросов к структуре. Затем в N строках следуют запросы по одному в строке: `GetMin`, `GetMax`, `Insert(A)` — извлечение минимума, максимума и добавление числа A ($1 \leqslant A \leqslant 2^{31} - 1$). Запросы корректны, то есть нет операций извлечения для пустого множества.

Формат выходных данных

Для каждого запроса `GetMin` или `GetMax` выведите то число, которое было извлечено.

Примеры

minmax.in	minmax.out
10 Insert(100) Insert(99) Insert(1) Insert(2) GetMin GetMax Insert(1) GetMin GetMin GetMax	1 100 1 2 99

Задача C. Англо-латинский словарь

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

Однажды, разбирая старые книги на чердаке, школьник Вася нашёл англо-латинский словарь. Английский он к тому времени знал в совершенстве, и его мечтой было изучить латынь. Поэтому попавшийся словарь был как раз кстати.

К сожалению, для полноценного изучения языка недостаточно только одного словаря: кроме англо-латинского необходим латинско-английский. За неимением лучшего он решил сделать второй словарь из первого.

Как известно, словарь состоит из переводимых слов, к каждому из которых приводится несколько слов-переводов. Для каждого латинского слова, встречающегося где-либо в словаре, Вася предлагает найти все его переводы (то есть все английские слова, для которых наше латинское встречалось в его списке переводов), и считать их и только их переводами этого латинского слова.

Помогите Васе выполнить работу по созданию латинско-английского словаря из англо-латинского.

Формат входных данных

Во входном файле содержатся несколько описаний английских слов. Каждое описание содержится в отдельной строке, в которой записано сначала английское слово, затем отделённый пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого английского слова на латинский. Переводы отсортированы в лексикографическом порядке. Порядок следования английских слов в словаре также лексикографический.

Все слова состоят только из маленьких латинских букв, длина каждого слова не превосходит 15 символов. Общее количество слов на входе не превышает 100 000.

Формат выходных данных

Программа должна вывести количество латинских слов в словаре k. В следующих k строках программа должна вывести латинско-английский словарь, соответствующий входному словарю, в точности соблюдая формат входных данных. В частности, первым должен идти перевод лексикографически минимального латинского слова, далее — второго в этом порядке и т. д. Внутри перевода английские слова должны быть также отсортированы лексикографически.

Примеры

dictionary.in	dictionary.out
apple - malum, pomum, popula fruit - baca, bacca, popum punishment - malum, multa	7 baca - fruit bacca - fruit malum - apple, punishment multa - punishment pomum - apple popula - apple popum - fruit

Задача D. Палиндром

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Палиндром — это строка, которая читается одинаково как справа налево, так и слева направо.

На вход программы поступает набор больших латинских букв (не обязательно различных). Разрешается переставлять буквы, а также удалять некоторые буквы. Требуется из данных букв по указанным правилам составить палиндром наибольшей длины, а если таких палиндромов несколько, то выбрать первый из них в алфавитном порядке.

Формат входных данных

Входные данные содержат одну непустую строку, состоящую лишь из не более чем 10^5 заглавных латинских символов, без пробелов.

Формат выходных данных

В единственной строке выходных данных выведите искомый палиндром.

Примеры

стандартный ввод	стандартный вывод
AAB	ABA
QAZQAZ	AQZZQA

Задача E. Электрички

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

На вокзале есть K тупиков, куда прибывают электрички. Этот вокзал является их конечной станцией, поэтому электрички, прибыв, некоторое время стоят на вокзале, а потом отправляются в новый рейс (в ту сторону, откуда прибыли).

Дано расписание движения, в котором указаны события прибытия и отбытия для каждой из электричек в хронологическом порядке. Поскольку вокзал — конечная станция, то электричка может стоять на нем довольно долго, в частности, электричка, которая прибывает раньше другой, отправляться обратно может значительно позднее.

Тупики пронумерованы числами от 1 до K. Когда электричка прибывает, ее ставят в свободный тупик с минимальным номером.

Напишите программу, которая по данному расписанию для каждой электрички определит номер тупика, куда прибудет эта электричка.

Формат входных данных

В первой строке вводится число K — количество тупиков ($1 \leqslant K \leqslant 20000$). Далее следуют строки, описывающие события прибытия/отбытия электричек. Каждая электричка задаётся своей противоположной конечной станцией — строкой длины не более 15 из латинских букв и знаков подчёркивания. Событие `+city` означает, что прибывает электричка из города `city`, событие `-city` — что эта электричка отправляется обратно. Общее количество электричек, фигурирующих в условии — не более 100000, для каждой фигурирующей электрички присутствуют оба события.

Считается, что в нулевой момент времени все тупики на вокзале свободны.

Формат выходных данных

Выведите по одному числу на каждую электричку — номер тупика, куда её поставят по прибытии. Если тупиков не достаточно для того, чтобы организовать движение электричек согласно расписанию, выведите число 0 и город первой из электричек, которая не сможет прибыть на вокзал.

Примеры

trains.in	trains.out
3 +bologoe +moscow -bologoe +stpetersburg -stpetersburg +samara +saratov -moscow -samara -saratov	bologoe 1 moscow 2 stpetersburg 1 samara 1 saratov 3
2 +kostroma +sudislavl +newvasyuki -sudislavl -kostroma -newvasyuki	0 newvasyuki

Задача F. Коньки

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

В ЛКШ-Зима школьники любят ходить на каток. В прокате коньков есть много коньков самых разных размеров. Школьник может надеть коньки любого размера, который не меньше размера его ноги. Известны размеры всех коньков и размеры ног школьников. Определите, какое наибольшее число школьников сможет одновременно пойти покататься.

Формат входных данных

Первая строка входных данных содержит число N — количество коньков в прокате ($1 \leqslant N \leqslant 10^5$). Во второй строке записано N чисел — размеры коньков. В третьей строке содержится число M — количество школьников в ЛКШ ($1 \leqslant M \leqslant 10^5$), четвертая строка содержит размеры их ног. Размеры коньков и ног — натуральные числа, не превосходящие 100.

Формат выходных данных

Выведите единственное число — наибольшее количество школьников, которое сможет пойти на каток.

Примеры

skates.in	skates.out
4 41 40 39 42 3 42 41 42	2

Задача G. Снеговики

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

Для того, чтобы слепить снеговика, необходимо три снежных кома разного размера. В вашем распоряжении есть n снежных комов, которые представляют собой шары с радиусами r_1, r_2, ..., r_n. Снеговика можно слепить из любых трех комов, радиусы которых попарно различны. Например, из комов с радиусами 1, 2 и 3 можно слепить снеговика, а из комов с радиусами 2, 2, 3 или 2, 2, 2 — нельзя. Определите, какое наибольшее количество снеговиков можно слепить из данных комов.

Формат входных данных

В первой строке входных данных задано целое число n ($1 \leqslant n \leqslant 10^5$) — количество комов. В следующей строке заданы n целых чисел — радиусы комов r_1, r_2, ..., r_n ($1 \leqslant r_i \leqslant 10^9$). Радиусы комов могут совпадать.

Формат выходных данных

В первой строке выведите одно целое число k — наибольшее количество снеговиков. Следующие k строк должны содержать описания снеговиков. Описание каждого снеговика должно состоять из трех чисел, разделенных пробелами — радиуса большого кома, радиуса среднего кома и радиуса маленького кома. Снеговиков разрешается выводить в любом порядке. Если решений несколько, выведите любое.

Примеры

snowmen.in	snowmen.out
7 1 2 3 4 5 6 7	2 7 6 5 4 3 2
3 2 2 3	0

Задача H. ACM Марафон

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

Школьник Вася Иванов так сильно боялся идти на командную олимпиаду, что ему приснился кошмар: в ЛКШ вместо обычной олимпиады устраивали обязательный ACM-марафон. Это почти обычная командная олимпиада, отличается она только продолжительностью. Марафон длится ровно 24 часа, то есть если он начался в 00:00:00 то в 23:59:59 команда еще может сдать решение, а в 00:00:00 следующего дня — уже нет.

Как и в обычном турнире ACM, побеждает команда, решившая наибольшее число задач, а при равном количестве решенных задач лучше результат у той команды, у которой меньше штрафное время. Изначально штрафное время каждой команды равно нулю. За каждую правильно сданную задачу к штрафному времени команды прибавляют время в минутах, округленное вниз, прошедшее с начала соревнования до момента сдачи задачи. Кроме того, если зачтённой попытке предшествовало несколько неудачных попыток сдать ту же задачу, то за каждую из них к штрафному времени прибавляют двадцать минут. За неудачные попытки сдать задачу, которую команде в итоге так и не удалось решить, штрафного времени не начисляется. Так же посылки с результатом “Compilation error” и “Code style violation” не считаются неудачными, то есть за них не начисляются штрафные минуты.

Вам требуется написать программу, которая подсчитает результаты марафона.

Формат входных данных

В первой строке входного файла находится время начала олимпиады в формате $hh : mm : ss$, где двухразрядное целое число hh ($0 \leqslant hh \leqslant 23$) означает час, а двухразрядные целые числа mm и ss ($0 \leqslant mm, ss \leqslant 59$) — минуты и секунды соответственно.

Во второй строке находится единственное целое число n ($1 \leqslant n \leqslant 1000$) — количество посылок за олимпиаду.

Далее следуют n строк с описаниями посылок. В начале каждой из них в двойных кавычках записано название команды, сделавшей посылку. Название может состоять из строчных и заглавных латинских букв, пробелов и цифр от 1 до 9. Длина названия — не меньше одного символа и не больше 255. После названия команды написано время посылки в том же формате, что и время начала контеста.

Далее через пробел идет заглавная латинская буква — номер задачи. Последние два символа в строке — результат посылки. Результат посылки может быть одни из следующих:
OK — OK
WA — Wrong answer
PE — Presentation error
TL — Time limit
ML — Memory limit
CE — Compillation error
CS — Code style violation

Формат выходных данных

Выходной файл должен содержать итоговую таблицу результатов — по строке на каждую команду. Строки должны идти в порядке уменьшения результата, если у нескольких команд результаты равны, то порядок команд определяется названием — раньше идет та, название которой лексикографически меньше.

Каждая строка должна начинаться с места команды в итоговом зачете. Место команды — это $k + 1$, где k — число команд, имеющих строго лучший результат. Далее через пробел идет название команды в двойных кавычках, а за ним через пробел два числа — количество решенных задач и штрафное время.

Примеры

contest.in	contest.out
00:00:00 5 "Super team" 00:00:23 A WA "Mega team" 00:10:21 A WA "Super team" 00:20:23 A OK "Mega team" 00:30:23 A OK "Mega team" 00:40:23 B OK	1 "Mega team" 2 90 2 "Super team" 1 40
01:00:00 3 "Team1" 01:10:00 A WA "Team1" 01:20:00 A OK "Team2" 01:40:00 B OK	1 "Team1" 1 40 1 "Team2" 1 40

Задача I. Постфиксная запись

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	1 second
Ограничение по памяти:	64 megabytes

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.

Дано выражение в обратной польской записи. Определите его значение.

Формат входных данных

В единственной строке записано выражение в постфиксной записи, содержащее однозначные числа и операции $+$, $-$, $*$. Строка содержит не более 100 чисел и операций.

Формат выходных данных

Необходимо вывести значение записанного выражения. Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений по модулю меньше 2^{31}.

Примеры

stdin	stdout
8 9 + 1 7 - *	-102

Задача J. Проверка ПСП

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дана строка, состоящая из круглых, квадратных и фигурных скобок. Нужно проверить, является ли она правильной скобочной последовательностью.

Формат входных данных

Во входном файле записана скобочная последовательность длиной не более 10 000 символов.

Формат выходных данных

Выведите `YES`, если скобочная последовательность является правильной, и `NO` в противном случае.

Примеры

stdin	stdout
`([]())`	`YES`
`([)]`	`NO`

Задача K. Марсианская парикмахерская

Имя входного файла:	`saloon.in`
Имя выходного файла:	`saloon.out`
Ограничение по времени:	1 second
Ограничение по памяти:	64 megabytes

Пока я не поспал, «сегодня» не наступило

мистер Грин

В парикмахерской работает один мастер. Он тратит на одного клиента ровно 20 минут, а затем сразу переходит к следующему, если в очереди кто-то есть, либо ожидает, когда придет следующий клиент.

Даны времена прихода клиентов в парикмахерскую (в том порядке, в котором они приходили).

Также у каждого клиента есть характеристика, называемая *степенью нетерпения*. Она показывает, сколько человек может максимально находиться в очереди перед клиентом, чтобы он дождался своей очереди и не ушел раньше. Если в момент прихода клиента в очереди находится больше людей, чем степень его нетерпения, то он решает не ждать своей очереди и уходит. Клиент, который обслуживается в данный момент, также считается находящимся в очереди.

Требуется для каждого клиента указать время его выхода из парикмахерской.

Формат входных данных

В первой строке вводится натуральное число N, не превышающее 10^5 — количество клиентов.

В следующих N строках вводятся времена прихода клиентов — по два числа, обозначающие часы и минуты (часы — от 0 до 16 000 000, минуты — от 0 до 59) и степень его нетерпения (неотрицательное целое число не большее 10^5) — максимальное количество человек, которое он готов ждать впереди себя в очереди. Времена указаны в порядке возрастания (все времена различны).

Если для каких-то клиентов время окончания обслуживания одного клиента и время прихода другого совпадают, то можно считать, что в начале заканчивается обслуживание первого клиента, а потом приходит второй клиент.

Формат выходных данных

В выходной файл выведите N пар чисел: времена выхода из парикмахерской 1-го, 2-го, ..., N-го клиента (часы и минуты). Если на момент прихода клиента человек в очереди больше, чем степень его нетерпения, то нужно считать, что время его ухода равно времени прихода.

Примеры

saloon.in	saloon.out
3 10 0 0 10 1 1 10 2 1	10 20 10 40 10 2
5 1 0 100 2 0 0 2 1 0 2 2 3 2 3 0	1 20 2 20 2 1 2 40 2 3

Задача L. Гоблины и шаманы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	64 мегабайта

Гоблины Мглистых гор очень любят 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 данный записано число N ($1 \leqslant N \leqslant 2 \cdot 10^5$) - количество запросов к программе. Следующие N строк содержат описание запросов в формате:

- «`+ i`» — гоблин с номером i ($1 \leqslant i \leqslant N$) 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 зайти к шаманам.

Примеры

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

Задача M. Имперский марш

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

На этот раз Император нагрянул с ревизией не в какой-то там ангар, а в казармы 501-го легиона имперских штурмовиков. В связи с этим каждого штурмовика постригли «под ежика». Несмотря на развитие нанотехнологий, постригли плохо — в результате из-за различной длины волос штурмовики могут отличаться друг от друга по росту, но незначительно — разница не превышает 137 нанометров. Ваша задача — выстроить штурмовиков по росту.

Формат входных данных

Первая строка входного файла содержит целое число N — количество штурмовиков ($1 \leqslant N \leqslant 100\,000$), вторая строка N — натуральных чисел, не превышающих $2 \cdot 10^9$ каждое — рост штурмовика в нанометрах. Никакие два роста не различаются более, чем на 137 нм.

Формат выходных данных

Выведите роста штурмовиков в порядке неубывания.

Примеры

march.in	march.out
5 12 1 2 1 13	1 1 2 12 13
1 1000000000	1000000000

Замечание

При решении этой задачи нельзя пользоваться стандартными функциями и методами `min`, `index`, `sort`, `sorted` и т. д.

Естественно, можно пользоваться функциями `min`, `max`, которые принимают два числа.