Задача A. Теоретический минимум

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

— Привет, Макс! — Алексей, как обычно, «поймал» Максима, когда тот уже собрался выключать компьютер. Конечно, Максим мог бы сказать потом, что уже засыпал и не заметил сообщения. Но утром экзамен, и почти наверняка Алексею нужно что-то по делу.
— Привет, Алекс!
— У меня есть отличная идея! Давай напишем игру! Надо только решить, под какую платформу писать — под Coriander или всё же под TinMan.
Да уж, дело... Несомненно, срочное. И чрезвычайно важное. Кстати, может, он зачёт не сдал, и на экзамен не собирается?
— У меня тоже отличная идея. Выспаться перед матанализом.
— А вот говорят, что голова лучше всего работает, когда слегка не выспался. Ты же всё выучил, получишь завтра свою «пятёрку». Так что сейчас расскажу, что придумал.
— Нет, давай завтра, а то усну на половине рассказа. Я подожду, пока ты ответишь, и обсудим твою идею, — Максим всегда держал слово. Впрочем, он был совершенно уверен, что к моменту обсуждения Алексея будет занимать уже какая-нибудь другая идея, — Ты хотя бы теоретический минимум выучил? Без него даже билет нельзя взять.
— А, так вот почему там два файла с вопросами было... У тебя же наверняка есть краткие ответы?
— Есть. Сейчас и у тебя будут. А я — спать, — Максим отправил файл и отключился.

Теоретический минимум состоит из n определений. Для каждого определения задано время t_j, в течение которого Алексей может его выучить. Однако, как только Алексей выучивает это определение, время, необходимое для выучивания всех последующих определений уменьшается на одну единицу времени. Разумеется, время не может быть отрицательным: если для какого-либо определения достигнуто нулевое значение, более оно не уменьшается.

В распоряжении Алексея m минут: спустя это время он уснёт. Он хочет успеть изучить как можно больше определений. Алексей читает определения в том порядке, в котором они записаны в конспекте Максима, возможно, пропуская некоторые.

Ваша задача — посчитать максимальное количество определений, которое сможет выучить Алексей.

Формат входного файла

В первой строке содержатся числа n и m ($1 \le n \le 100000, 1 \le m \le 10^{10}$) — количество определений в конспекте и количество минут в распоряжении Алексея.

Во второй строке содержится n целых чисел t_j ($j = 1, 2, \ldots, n,\ 1 \le t_j \le 100000$) — время, необходимое, чтобы выучить соответствующее определение.

Формат выходного файла

Выведите единственное целое число — максимально возможное количество определений, которое Алексей сможет выучить

Примеры

input.txt	output.txt
7 20 5 12 3 7 6 8 11	5
6 12 12 2 2 2 12 4	5

Задача B. Отличная идея

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Алексей — человек увлекающийся. К примеру, приходит ему в голову идея написать игру, действие которой происходит в другой галактике. И он начинает придумывать, как могут выглядеть межзвёздные корабли, на которых будут перемещаться персонажи. Потом он принимается рассуждать о том, из какого материала может быть изготовлена обшивка корабля и какую нужно накладывать текстуру на модель. Затем его посещает мысль поискать в интернете текстуру, которую можно взять за основу. Обнаружив неплохую библиотеку, он заодно просматривает текстуры ландшафта, задумывается, как мог бы выглядеть инопланетный животный мир... Словом, ничего удивительного, если спустя несколько часов он готов рассказывать уже не об игре, а о разработке самовосстанавливающихся защитных покрытий методами генной инженерии. Впрочем, он может в этот момент ещё помнить и про задуманную игру, и про обшивку корабля, и про животный мир.

В течение ночи Алексей обдумал n идей. Для каждой j-ой идеи известны два момента времени s_j и f_j ($s_j < f_j$): в момент s_j эта идея заинтересовала Алексея, а в момент f_j он утратил к ней интерес. Алексей может обдумывать эту идею в течение некоторого промежутка времени между s_j и f_j. Более того, он может переходить к какой-то другой идее — из тех, которые к этому моменту уже пришли ему в голову, а затем возвращаться к какой-то из уже обдуманных. Алексей помнит про каждую идею, начиная с того момента, в который он начал ее обдумывать, и в течение всего времени, пока она его интересует.

Если Алексей начал обдумывать какую-то идею (неважно, в который раз), то он потратит на это по меньшей мере одну единицу времени.

Назовём глубиной контекста количество идей, о которых одновременно помнит Алексей. Ваша задача — по заданным моментам времени определить максимальную глубину контекста в эти моменты.

Формат входного файла

В первой строке записаны целые числа n и q ($1 \le n \le 100000, 1 \le q \le 100000$) — количество идей, которые обдумал Алексей, и количество моментов времени, для которых нужно найти максимальную глубину контекста.

В каждой из следующих n строк записано по два целых числа s_j и f_j ($0 \le s_j < f_j \le 10^9$) — целые числа, моменты времени, в которые идея j занимала внимание Алексея.

Далее, в каждой из следующих q строк записано по одному целому числу t_i ($i = 1, 2, \ldots, q$, $\min\limits_{j=1..n}\{s_j\} \le t_i \le \max\limits_{j=1..n}\{f_j\}$) — моменту времени, для которого нужно найти максимальную глубину контекста.

Формат выходного файла

Выведите q строк, в каждой из которых содержится одно целое число — максимальная глубина контекста в соответствующий момент времени.

Примеры

input.txt	output.txt
7 10 2 4 4 6 1 5 8 12 3 5 11 12 2 3 4 1 2 6 5 9 3 11 7 2	4 1 2 1 3 1 3 2 0 2

Задача C. Поверхностный интеграл

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Алексей уже не один раз пожалел, что, не взглянув на задачу, стал готовить ответы на вопросы билета. С билетом ему повезло — оба вопроса он неплохо знал. Но вот задача...

Интегралы по поверхностям были последней темой в семестре. А последние страницы конспекта Алексей пролистывал уже под утро. Да и заучить теорему совсем не значит уметь применить её в нетривиальном случае. Случай был явно нетривиальным — преподаватель снабдил задачу рисунком, что бывает крайне редко. И самое грустное — когда Алексей посмотрел на задачу, уже подходила очередь Максима отвечать.

Максим попытался ему помочь, оставил набросок решения, пестревший многоточиями и пометками «досчитать до числа». И теперь Алексей с грустью смотрел на эти формулы — даже если переписать самое начало, где вроде бы нет пропусков, получится ли это объяснить?..

Картинка с интегралом, добросовестно перерисованная Алексеем на листок, постепенно превращалась в эскиз космической станции из придуманной игры. Ну вот зачем будущему разработчику игр уметь считать интегралы по поверхностям? Хотя, наверное, зачем-то надо. Можно Максу сказать, что это бесполезное знание — и он за пару минут придумает, почему Алексею без этого знания ну совершенно никак нельзя.

Нет, вообще зачем он, Алексей, пошел следом за Максимом на математику? Макс тут как рыба в воде, ему всё легко даётся, те же интегралы — сразу видит, какую замену или подстановку сделать. Да ещё и говорит, что похожи все эти интегралы. Для Алексея они все совершенно разные. Макс, конечно, утверждает, что если бы Алексей делал домашние задания и не прогуливал лекции, то всё было бы по-другому. Зато Макс меньше языков программирования знает. Хотя...

Да, как ни грустно признавать это Алексею, за последние полгода Макс стал программировать намного лучше. И иной раз может такие вещи рассказать про особенности компиляторов или виртуальных машин, о которых Алексей даже и не задумывался. А ещё он на стажировку летом куда-то собрался. Так что скоро никаких «зато» у Алексея и не останется. Ладно, со следующего семестра...

Алексей переписал те фрагменты из записей Максима, в которых разобрался, и машинально перевернул листок. Ассемблер?.. Нет, похоже, да не совсем. Интересно, чем это Макс занялся? Неужели написал-таки свой компилятор? И как тут думать про всякие интегралы по поверхностям?

Алексей решил использовать текст этой программы, чтобы «угадать», сдаст ли он экзамен. Он придумал следующую формулу (все величины считаются по тексту программы):

$$\begin{aligned}&\langle\textit{результат}\rangle = \langle\textit{количество различных букв}\rangle \cdot \langle\textit{количество чисел}\rangle - \\ &- \langle\textit{количество строк}\rangle \cdot \langle\textit{количество слов }\mathbf{def}\rangle - \langle\textit{количество слов }\mathbf{jmp}\rangle - \\ &-(\langle\textit{количество букв }\mathbf{m}\rangle + \langle\textit{количество букв }\mathbf{i}\rangle + \langle\textit{количество букв }\mathbf{n}\rangle) - \\ &-\langle\textit{количество слов }\mathbf{jg}\rangle \cdot (\langle\textit{количество букв }\mathbf{a}\rangle + \langle\textit{количество букв }\mathbf{z}\rangle)\end{aligned}$$

Если Алексей получает положительное число, он будет считать, что ему удастся получить положительную оценку. В противном случае ему придётся пойти на пересдачу.

Ваша задача — определить по тексту программы, какое число получит Алексей.

Формат входного файла

Программа, записанная на неизвестном Алексею языке программирования. В программе могут встречаться слова, записанные строчными латинскими буквами (каждое длиной не более 20 символов), и целые положительные числа (каждое из не более 20 цифр).

Программа содержит по меньшей мере одно слово. Пустые строки в программе не считаются. Всего слов и чисел в программе не более 100000.

Формат выходного файла

В первой строке выведите единственное целое число, которое получит Алексей при вычислениях по своей формуле.

Примеры

input.txt	output.txt
def my inc 10 jl 45 1 end def main call my call f end def f dec 2 jg 45 jmp 5 inc 8 dec 2 end	42
def z inc 2 jg 1 end def main call z end	-1

Задача D. Сумма технологий

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Наконец, Алексей вышел из аудитории, сообщив, что сегодня ему сопутствовала удача. Максим предполагал, что Алексей уже и не вспомнит о ночной задумке, однако Алексей весьма живо стал рассказывать ему об игре.

Действие игры происходит в далекой галактике. В результате столкновения этой галактики с другой расширяющееся кольцо активного звёздообразования подобно волне надвигается в том числе на небольшую звезду с планетной системой. Цивилизация, населяющая одну из планет, намерена предпринять усилия по собственному спасению.

Две группы учёных (назовём их группой **A** и группой **B**) предложили альтернативные варианты спасения. Не вдаваясь в подробности, скажем, что одна группа предлагает эвакуировать в безопасную зону всё население, другая же предлагает эвакуацию населения вместе с планетой.

Обе группы представили детальные планы по разработке технологий, позволяющих осуществить предлагаемые варианты эвакуации.

Для простоты будем считать, что каждый такой план состоит из описания n технологий, которые должны быть последовательно разработаны, и базовой стоимости разработки каждой технологии: для первой группы это значения $a_1, a_2, \ldots, a_n$, для второй — $b_1, b_2, \ldots, b_n$.

Однако, оценивая базовую стоимость разработки, каждая группа учёных рассчитывала, что будет частично пользоваться результатами другой группы. Правительство же в стремлении сэкономить поручило группе экономистов произвести расчёт финансирования для каждой из научных групп по отдельности. В результате этого анализа экономисты сообщили, что

- ◇ если в текущий момент разработаны технологии A_j и B_j, то разработка технологии A_{j+1} или B_{j+1} обойдётся в базовую стоимость — a_{j+1} или b_{j+1} соответственно;

- ◇ если в текущий момент разработаны технологии A_j и B_{j+1}, то разработка технологии A_{j+1} обойдётся в базовую стоимость a_{j+1}, а разработка технологии B_{j+2} — вдвое дороже базовой стоимости, т.е. в $2 \cdot b_{j+2}$;

  верно и обратное: если в текущий момент разработаны технологии A_{j+1} и B_j, то разработка технологии B_{j+1} обойдётся в базовую стоимость b_{j+1}, а разработка технологии A_{j+2} — вдвое дороже базовой стоимости, т.е. в $2 \cdot a_{j+2}$;

- ◇ если в текущий момент разработаны технологии A_j и B_{j+2}, то разработка технологии A_{j+1} обойдётся вдвое дешевле базовой стоимости — в $a_{j+1}/2$, а разработка технологии B_{j+3} невозможна, пока не будет ликвидировано технологическое отставание.

  верно и обратное: если в текущий момент разработаны технологии A_{j+2} и B_j, то разработка технологии B_{j+1} обойдётся вдвое дешевле базовой стоимости — в $b_{j+1}/2$, а разработка технологии A_{j+3} невозможна, пока не будет ликвидировано технологическое отставание.

Теперь правительство хочет получить план финансирования научных групп, который позволит потратить как можно меньше денег. Одновременно оно будет финансировать только одну научную группу. Как только одна из групп выполнит свой план по разработке технологий, правительство будет считать, что эвакуация стала возможной, и прекратит финансирование.

Ваша задача — определить сумму, которая будет истрачена на разработку технологий, а также — какой из планов эвакуации будет реализован.

Формат входного файла

В первой строке содержится целое число n ($1 \le n \le 100000$) — количество технологий в плане каждой научной группы.

Во второй строке содержится n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \le a_j \le 10000$, $j = 1, 2, \ldots, n$) — базовые стоимости разработки технологий научной группой **A**.

Во третьей строке содержится n целых чисел $b_1, b_2, \ldots, b_n$ ($1 \le b_j \le 10000$, $j = 1, 2, \ldots, n$) — базовые стоимости разработки технологий научной группой **B**.

Формат выходного файла

В первой строке выведите целое число — минимально возможную сумму (при необходимости округлённую вверх), которую правительство истратит на разработку технологий.

Во второй строке выведите слово POPULATION, если план будет полностью выполнен первой научной группой и PLANET, если план будет полностью выполнен второй научной группой. Если же возможны оба варианта, выведите RANDOM.

Примеры

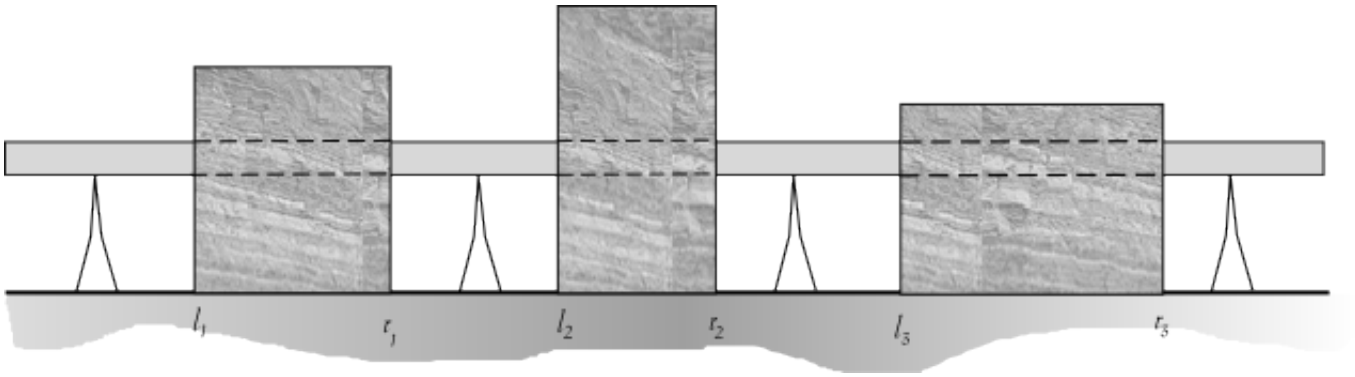
input.txt	output.txt
2 3 1 2 2	5 POPULATION
3 1 1 3 3 2 2	9 PLANET

Задача E. Тоннель

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Для одного из исследований научной группе **B** требуется построить линейный ускоритель. Чтобы исключить влияние внешних факторов, ускоритель должен располагаться в тоннеле длины L со специальной защитой. Такую защиту можно обеспечить либо «естественным путём», если тоннель будет окружён базальтовыми породами, либо «искусственным путём» — с помощью специальных инженерных сооружений.

Местность, в которой планируется построить ускоритель, расположена в трапповой провинции (образованной базальтовыми породами). Рельеф местности можно смоделировать набором непересекающихся прямоугольников (траппов) различной высоты. Одна сторона каждого из этих прямоугольников полностью лежит на оси абсцисс.



Начало и конец тоннеля должны быть расположены вне траппа (или на его границе).

Стоимость прокладки единицы длины тоннеля в базальтовой породе составляет b, а стоимость защиты единицы длины тоннеля с помощью специальных инженерных сооружений зависит от высоты над поверхностью земли и определяется как $c+k\cdot h$, где c — стоимость инженерных сооружений, расположенных на поверхности земли, h — расстояние до поверхности земли, k — коэффициент удорожания.

Ваша задача — определить, на какой высоте над поверхностью земли следует расположить тоннель, чтобы его сооружение обошлось как можно дешевле. Гарантируется, что построить тоннель всегда возможно.

Формат входного файла

В первой строке содержатся целые числа t ($1 \le t \le 10^5$) и n ($1 \le n \le 1000$) — требуемая длина тоннеля и количество траппов.

Во второй строке — целые числа b, c, k $1 \le b, c \le 1000$, $1 \le k \le 20$), определяющие стоимость прокладки тоннеля в разных условиях.

В третьей строке содержатся целые числа s и f ($-10^6 \le s < f \le 10^6$) — границы местности, доступной для сооружения ускорителя.

В каждой из следующих n строк содержится по три целых числа: l_j, r_j, h_j ($s \le l_j < r_j \le f, 1 \le h_j \le 1000, j = 1, 2, \ldots, n$) — значения левой и правой границ прямоугольника, отмеченные на оси абсцисс, а также его высота.

Прямоугольники заданы в порядке возрастания значений их границ.

Формат выходного файла

В первой строке — вещественное число c с абсолютной или относительной точностью не менее 9 знаков после запятой, стоимость сооружения тоннеля.

Примеры

input.txt	output.txt
25 2 20 2 1 -5 30 0 10 20 15 20 5	305
25 2 2 20 10 -5 35 -5 10 40 10 35 30	50

Задача F. Материаловедение

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Научная группа **A** выдвинула предположение, что экранировка тоннеля может быть значительно удешевлена при использовании материала, который эта группа недавно разработала.

Для изготовления этого материала используется прямоугольная сетка размера $n \times m$, состоящая из квадратных ячеек с единичной стороной. Чтобы запустить процесс формирования материала, в некоторые ячейки помещают так называемые «точки роста».

В ячейке будет сформирован материал,

— если в ней находится «точка роста»;

— если хотя бы две соседних с рассматриваемой ячейки заполнены материалом.

Ячейки считаются соседними, если граничат по стороне.

Ваша задача — определить по заданному расположению «точек роста» в каких ячейках сетки сформируется материал.

Формат входного файла

В первой строке содержатся целые числа n, m ($1 \le n, m \le 1000$) — размеры прямоугольной сетки.

В каждой из следующих n строк содержится по m заглавных латинских букв '`O`' или '`X`', где '`X`' обозначает «точку роста», а '`O`' — пустую ячейку.

Формат выходного файла

Выведите n строк, содержащих по m заглавных латинских символов '`O`' или '`M`', где '`M`' обозначает заполненную материалом ячейку, а '`O`' — пустую ячейку.

Примеры

input.txt	output.txt
4 4 OOOO OXOO OOXO OOOO	OOOO OMMO OMMO OOOO
5 6 OXOXOO XOOOXO OOOOOO XOOOOO OOXOOO	MMMMMO MMMMMO MMMMMO MMMMMO MMMMMO

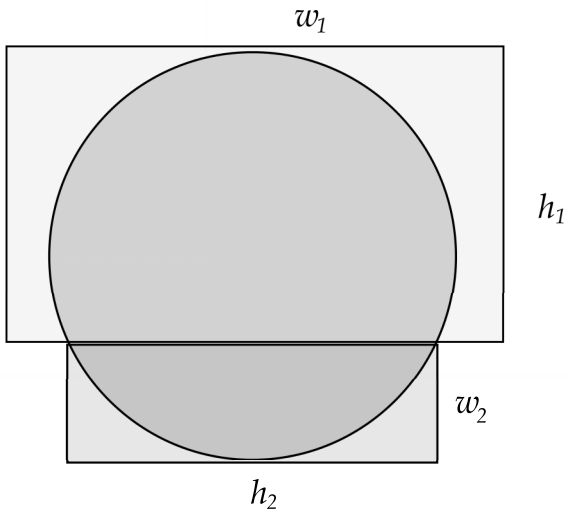
Задача G. Ловушка

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Для исследования частиц в ускорителе необходимо изготовить ловушки. Ловушки изготавливаются из материала со специальной структурой, имеют цилиндрическую форму, а в поперечном сечении — форму круга.

Особенности материала таковы, что менять направление оси симметрии цилиндров нельзя, а основания цилиндров должны быть перпендикулярны оси симметрии. Поэтому дальнейшее рассмотрение мы будем вести на плоскости.

Заготовки для ловушек в поперечном сечении имеют форму прямоугольника. При изготовлении ловушки допускается не более одной «стыковочной линии», параллельной диаметру круга. Из-за структурных особенностей материала «стыковочная линия» должна быть образована границами заготовок.



В распоряжении учёных есть две заготовки с различными поперечными сечениями. Ваша задача — определить максимально возможную площадь поперечного сечения ловушки, которую они смогут изготовить.

Формат входного файла

В первой строке содержится целое число n ($1 \le n \le 100$) — количество пар заготовок.

В каждой из следующих n строк содержится по четыре целых числа $w_1^{(j)}$, $h_1^{(j)}$, $w_2^{(j)}$, $h_2^{(j)}$ ($1 \le w_1^{(j)}, h_1^{(j)}, w_2^{(j)}, h_2^{(j)} \le 1000,\ j = 1, 2, \ldots, n$) — размеры заготовок.

Формат выходного файла

Выведите n строк: в каждой — вещественное число с абсолютной или относительной точностью не менее 6 знаков после запятой, максимально возможная площадь поперечного сечения ловушки.

Примеры

input.txt	output.txt
2 4 3 1 5 2 3 3 2	12.566370614359172463 99 7.068583470577034511 00

Задача H. Цветовой код

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Алексей вдохновенно рассказывал о том, как цивилизация на далекой планете боролась за выживание, как покинула свою галактику...

— Алекс, ты точно уверен, что рассказываешь мне про игру именно для мобильной платформы?

— Да, я вот как раз хотел сказать, что нынешние мобильные платформы несовершенны, их надо кардинально перерабатывать. Они всё ещё ориентированы на телефон, хотя очевидно же, что будущее за носимой электроникой и дополненной реальностью.

Алексей заговорил о том, что, к примеру, иконки или бегущая строка на каком-нибудь смартбраслете совершенно неудобны, и что можно было бы использовать цветовое меню — сопоставить разным функциям разные цветовые последовательности.

— Полоска из семи основных цветов, пальцев пять, получается, что при наборе цветового кода в худшем случае придется смещать палец на одну клетку влево или вправо, — рассуждал он.

Алексей полагает, что цвета в полоске должны подстраиваться под наиболее часто набираемые цветовые последовательности. Ему представляется, что проще всего безошибочно набрать код в случае, если просто скользить одним пальцем по браслету при наборе. Если же вдруг так не получается — то он хотел бы, чтобы количество перемещений пальца через одну или большее количество клеток было наименьшим. Такие перемещения Алексей для краткости называет «прыжками».

Алексей уже придумал n цветовых кодов, которые, по его мнению, являются легко запоминающимися. Теперь Алексей хотел бы узнать, в какой последовательности нужно разместить цвета в полоске на браслете, чтобы минимизировать количество «прыжков» для данного набора цветовых кодов. Переход от одного кода к другому коду не считается «прыжком».

Для обозначения цветов будем использовать заглавные буквы латинского алфавита: R — красный, O — оранжевый, Y — жёлтый, G — зелёный, B — голубой, I — синий, V — фиолетовый.

Формат входного файла

В первой строке содержатся целое число n ($1 \le n \le 1000$) — количество придуманных цветовых кодов.

В каждой из следующих n строк содержится по одному цветовому коду — непустой строке из символов, обозначающих цвета согласно описанию в условии. Суммарная длина всех строк не превышает 10000 символов.

Формат выходного файла

Выведите единственную строку из семи символов, обозначающих цвета, в том порядке, в котором они должны быть расположены на полоске.

Если ответов несколько, выведите любой.

Примеры

input.txt	output.txt
3 RGB BIV VOY	RGBIVOY
3 BOROV ROY YORIG	BVYORIG

Задача I. Каждый охотник желает знать...

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Когда Алексей придумывал цветовые коды, он, конечно, опирался на свои ассоциации — к примеру, вспоминал, как выглядит меню в кафе, или цветовую гамму, в которой оформлены вывески торговой сети...

И теперь его заинтересовал вопрос — а какие коды среди придуманных им используют не все цвета?

Алексей придумал n цветовых кодов, каждый из которых является непустой строкой, состоящей из не более чем 255 символов, обозначающих цвета. Для обозначения цветов используются заглавные буквы латинского алфавита: R — красный, O — оранжевый, Y — жёлтый, G — зелёный, B — голубой, I — синий, V — фиолетовый.

Ваша задача — определить, в каких кодах ни разу не встречается хотя бы один цвет.

Формат входного файла

В первой строке содержится целое число n ($1 \le n \le 1000$) — количество цветовых кодов, придуманных Алексеем.

В каждой из следующих n строк содержится по одному цветовому коду — непустой строке, состоящей из не более чем 255 символов, обозначающих цвета.

Формат выходного файла

В первой строке выведите через пробел в порядке возрастания номера кодов, в которых использованы не все цвета.

Если таких кодов нет, выведите 0.

Примеры

input.txt	output.txt
5 OIOBGR YRGGIVOOYB GRIG BOY BORGIVVROY	1 3 4
2 VIBGYOR RROGBIOORBYV	0

Задача J. Прицельная точность

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Максим слушал Алексея и думал, что цветовое меню может оказаться довольно удобной штукой: цветные кнопки будут достаточно крупными, нажать их правильно получится даже на ходу. Ведь чтобы набрать что-нибудь на экранной клавиатуре, приходится как минимум замедлять шаг. А уж когда опечатался, поставить курсор на неверно набранную букву в слове с первого раза может и не получиться.

Впрочем, Максима огорчало даже не это. Конечно, когда допущена опечатка, текстовый редактор смартфона выдает подсказки, но эти подсказки не кажутся Максиму такими уж полезными. Он уже не раз задумывался о том, как можно было бы усовершенствовать эту систему, и вот сейчас Алексей, вряд ли того желая, вернул его к этим размышлениям.

Максим считает, что пользователи набирают слова с использованием только строчных букв латинского алфавита, а набранный текст состоит из некоторого числа таких слов, разделённых произвольным количеством пробелов. Чтобы исправить слово, нужно сначала его выделить. Для этого нужно нажать на какую-нибудь букву этого слова.

Рассуждая так, Максим пришёл к следующей несложной задаче: он хотел бы научиться определять границы слова, когда пользователь нажимает на какую-нибудь букву этого слова. Границы слова — это номера первой и последней буквы этого слова.

Формат входного файла

В первой строке содержится текст, набранный пользователем — непустая строка s длиной m символов ($1 \le m \le 100000$), состоящая из строчных латинских букв и пробелов. Она не начинается на пробел и не заканчивается пробелом.

Во второй строке содержится целое число n ($1 \le n \le 100000$) — количество запросов.

В каждой из следующих n строк содержится по одному целому числу q_j ($j = 1, 2, \ldots, n,\ 1 \le q_j \le m$) — номер символа, на который нажал пользователь.

Формат выходного файла

Выведите n строк — по одной строке в ответ на каждый запрос.

В каждой строке должны содержаться по два целых числа l_j и r_j ($j = 1, 2, \ldots, n$) через пробел — номер символа, с которого начинается слово, содержащее нажатый пользователем символ, и номер символа, которым это слово заканчивается (нумерация ведётся с единицы).

Если пользователь нажал на пробельный символ, следует вывести в качестве l_j и r_j номер этого символа.

Примеры

input.txt	output.txt
hello world 4 2 7 12 2	1 5 7 7 9 13 1 5

Задача K. Следите за котиками!

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Максим добрался домой и, наконец, погрузился в обдумывание системы подсказок. Он заварил себе чаю, открыл большую упаковку печенья...

Печенье, которое нравится Максиму, имеет вид фигурок разных животных: есть там и зайцы, и обезьяны, и собаки, и верблюды, и, конечно же, котики. Увлёкшись, Максим может съесть хоть всю упаковку. И чтобы этого не случилось, он поступает следующим образом.

Максим ест печенье за печеньем, пока не вытащит печенье в виде котика. Он запоминает, что уже съел такое печенье и, когда ему вновь попадается печенье в виде котика, он возвращает его обратно, считая, что уже достаточно съел печенья на сегодня.

Будем обозначать печенье в виде котика заглавной латинской буквой C, а печенье в виде любого другого животного — буквой V. В Вашем распоряжении имеется строка, описывающая последовательность печений в том порядке, в каком их доставал Максим.

Ваша задача — определить, какое максимальное количество печений за день Максим мог съесть, соблюдая свои правила.

Формат входного файла

В первой строке содержится строка из латинских букв C и V длиной не более 100000 символов.

Формат выходного файла

В первой строке выведите целое число — максимальное количество печений, которое Максим мог съесть подряд.

Примеры

input.txt	output.txt
VVVCVVVVCVVVVCVVVVVCVCVCC	10
VCCVCCC	2

Note

Пояснение к первому примеру.

В первый день Максим съедает 8 печений: 9-ое печенье в виде котика он уже не ест, возвращает в упаковку. Во второй день он съедает 10 печений: сначала 4 в виде разных животных, затем одно в виде котика, затем еще 5 в виде разных животных, и, наконец, вновь вытаскивает печенье в виде котика (которое он возвращает в упаковку). В третий день он съедает сначала печенье в виде какого-то животного, затем печенье в виде котика, затем снова печенье в виде какого-то животного, и снова вытаскивает печенье в виде котика, которое приходится положить обратно. Таким образом, в третий день он съест всего 3 печенья. В четвертый день он вытащит из упаковки единственное оставшееся печенье в виде котика, съест его, и упаковка закончится.

Максимальное количество печений, которое Максим съел за один день, составляет 10.