



A. Big Chat

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Участники некоего форума хотят организовать между собой чат таким образом, чтобы одновременно в нем оказалось максимальное число посетителей. Для каждого участника известны интервалы времени, когда он может выйти на связь. Число посетителей, одновременно находящихся в чате, подсчитывается каждую минуту независимо от их состава.

Нужно найти это максимальное значение и максимальную длительность такого чата (число минут) в пределах календарных суток, то есть при переходе через полночь длительности подсчитываются отдельно в начале и в конце суток.

Входные данные

В первой строке дано число N ($2 \leq N \leq 50$) — количество участников. Далее следует N строк с интервалами доступности участника.

В начале каждой строки находится число интервалов I ($1 \leq I \leq 720$). Далее даны через пробелы $2 \times I$ строк — интервалы времени суток с градацией в 1 минуту и граничными значениями от 00час00мин до 23час59мин: начало 1-го интервала, конец 1-го интервала, затем — начало 2-го интервала и так далее. Границы интервалов задаются в формате hh:mm, где hh — число часов (от 0 до 23), а mm — число минут (от 0 до 59). Если интервал переходит через полночь, то его под-интервал после полуночи относится к началу тех же суток. Конец интервала означает, что в данное время участник уже вышел из чата.

Выходные данные

Выведите 2 числа через пробел: максимальное количество посетителей, одновременно находящихся в чате, и максимальную продолжительность такого чата в пределах календарных суток (количество минут).

Примеры

Входные данные	Результат	Пояснение
2 1 00:07 00:13 1 00:09 00:11	2 2	
2 2 01:07 01:13 23:00 00:12 1 00:09 01:11	2 4	Здесь 2 посетителя и время с 01:07 до 01:11
2 1 00:00 23:59 1 23:59 00:00	1 1440	Здесь количество посетителей 1 в интервале времени суммарной длиной 1 сутки
3 2 00:14 00:15 23:00 00:12 1 23:12 00:11 1 23:59 00:14	3 11	Здесь для числа посетителей 3 максимальный интервал времени с 00:00 до 00:11



В. Федя и путешествие

Ограничение по времени: для C и C++ — 2 секунды, для Java — 5 секунд
Ограничение по памяти: 256МБ

Наконец наступили долгожданные каникулы и Федя полон энтузиазма отправиться в путешествие — на лавовое поле Оудаудахрёйн, которое находится в столице Берляндии. Помимо столицы, Федор также хочет побывать на любой окраине Берляндии - в любом городе, из которого есть не больше одной дороги. Для этого Федор хочет выбрать путь, начинающийся на окраине Берляндии и заканчивающийся в столице. Гарантируется, что в Берляндии можно добраться из любого города в любой и одним путем.

В каждом городе метро принадлежит какой-то компании. Одна компания может быть владельцем метро в нескольких городах. Чтобы перемещаться по городу, Федору необходимо купить пропуск в метро для каждого города, который он собирается посетить. Один раз купив пропуск какой-то компании, Федор может его использовать в любом городе, где эта компания владеет метро. Помогите Федору найти такой маршрут, при котором суммарная стоимость пропусков была бы как можно меньше.

Входные данные

В первой строке находится два числа N и M ($1 \leq M \leq N \leq 10^5$) — количество городов и количество компаний.

В следующих N строках записаны пары: число P_i ($0 \leq P_i \leq N$) и строка C_i (от 1 до 10 маленьких английских букв), где P_i — город, с которым соединен город i , а C_i — название компании, которая владеет метро в этом городе. Для столицы $P_i=0$. Гарантируется, что из любого города можно попасть в любой другой и по единственному пути. Гарантируется, что есть ровно один город, для которого $P_i=0$.

Далее следует M строк, описывающих компании: название компании и стоимость пропуска для проезда на метро, принадлежащем этой компании (целое число от 0 до 1000).

Выходные данные

Выведите одно число — минимальную возможную суммарную стоимость пропусков на маршруте Федора.



Примеры

Входные данные	Результат
3 3 0 mapple 1 moomle 2 minisoft mapple 100 moomle 200 minisoft 300	600
6 4 5 cocacola 3 snapchat 0 snapchat 2 facepen 3 ibm 3 snapchat snapchat 994 cocacola 434 ibm 702 facepen 186	994



С. Чай с печенюшками

Ограничение по времени: для С и С++ — 1 секунда, для Java — 2 секунды
Ограничение по памяти: 256МБ

Что может быть лучше, чем выпить чашечку чая с лучшей подругой под вкуснейшее печенье? Но вот незадача — на столе стоит N тарелок с печеньем, каждая из которых содержит A_i печенюшек. Вам с подругой необходимо разделить печенье поровну, чтобы никто не остался в обиде.

Определите, возможно ли разделить имеющиеся печенье поровну между вами и подругой.



Входные данные

В первой строке задано число N ($1 \leq N \leq 10^3$) — количество тарелок на столе. В последующих N строках задано количество печенюшек в i -ой тарелке cnt_i ($1 \leq cnt_i \leq 10^{10000}$).

Выходные данные

В единственной строке выведите YES, если печенюшки можно разделить поровну, иначе — NO.

Примеры

Входные данные	Результат
3 12 11 5	YES
2 6 3	NO



D. Мобильный интернет

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

В этом году Вася Пупкин, студент X-OSPU (eXtreme Open Source Programming University), уже четверокурсник. А что это значит? Правильно, это значит, что он стал ещё меньше ходить на пары. Да, так мало, что с некоторыми преподавателями он знакомится прямо на... Нет, что вы, не на экзамене. На модульном контроле.

Так как же Васе написать контрольную по предмету «Экономика Разработки Усовершенствований, Направленных На Дополнительную Автоматизацию» («ЭРУННДА»), если он не успел выучить даже название предмета? Мобильный интернет приходит на помощь! Вася, пряча смартфон под партой, уже успел наугадить два подходящих конспекта. Но есть проблема: конспекты, похоже, сделаны картинками, поэтому файлы — большие. А с тех пор, как Вася скачал обновление для своего смартфона до RobotOS 8.1 Octopus через своего оператора deathcell:(, у него каждый мебибайтик на счету. «Скачаю тот, что меньше», — решил Вася.

Но сравнить размеры файлов оказалось не так-то просто. На одном сайте размер указан в мегабайтах, а на другом — в мебибайтах. При этом сами числа близкие, так что без точных подсчётов не обойтись. К сожалению, после обновления калькулятор у Васи на смартфоне почему-то всё время падает, поэтому Вася просит вас помочь с расчётами.

Вам даны размеры обоих файлов. Определите, какой из них меньше, либо что они равны.

Входные данные

В первой строке указан размер первого файла, а во второй — размер второго файла. Оба размера заданы в формате «целое число, пробел, единица измерения». Единица измерения может быть B, KB, KiB, MB, MiB, GB, GiB, TB, TiB, PB, PiB, EB или EiB (смотрите таблицу на следующей странице). Размеры каждого из файлов строго меньше 8 эксбибайт.

Выходные данные

Выведите один символ — знак отношения, который нужно поставить между размером первого файла и размером второго файла: <, > или =.

Примеры

Входные данные	Результат
42 MB 41 MiB	<
1000 KiB 1024 KB	=
41 MiB 42991616 B	=



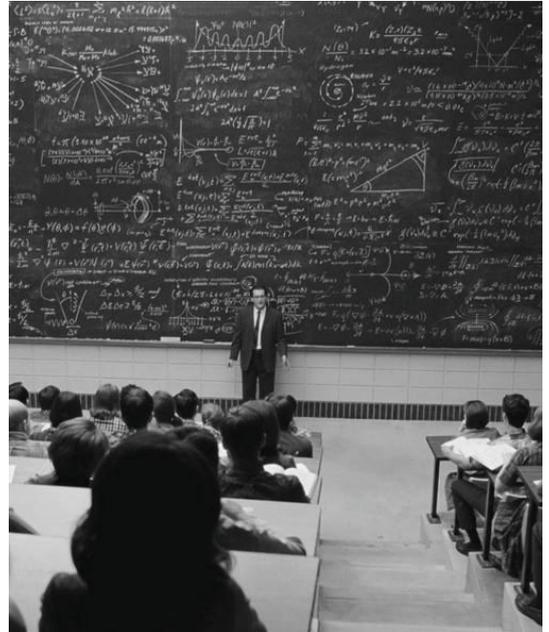
Справочная информация

Приставка	Множитель	
K (kilo, кило)	10^3	1 000
Ki (kibi, киби)	2^{10}	1 024
M (mega, мега)	10^6	1 000 000
Mi (mebi, меби)	2^{20}	1 048 576
G (giga, гига)	10^9	1 000 000 000
Gi (gibi, гиби)	2^{30}	1 073 741 824
T (tera, тера)	10^{12}	1 000 000 000 000
Ti (tebi, теби)	2^{40}	1 099 511 627 776
P (peta, пета)	10^{15}	1 000 000 000 000 000
Pi (pebi, пеби)	2^{50}	1 125 899 906 842 624
E (exa, экса)	10^{18}	1 000 000 000 000 000 000
Ei (exbi, эксби)	2^{60}	1 152 921 504 606 846 976

Е. И снова математика

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

На очередной паре по высшей математике не оказалось мела в аудитории. Преподавателю необходимо выбрать, кто же спустится с 9-го этажа за мелом. Но желающих не оказалось. И тут преподаватель решил схитрить и предложить одному из студентов пари: «Берем случайное N -значное десятичное число и, если там оказываются две одинаковые цифры подряд, то за мелом иду я, иначе — вы». Студент Коля не дурак и не согласится на такое пари, если вероятность его выигрыша не будет хотя бы 99%. Коле очень важно не допустить ошибки, так как искушение выиграть спор у преподавателя очень велико! Но поскольку Коля не очень силен в математике, вам необходимо помочь ему узнать, стоит ли браться за это пари.



Входные данные

В единственной строке содержится число N ($1 \leq N \leq 10^9$) — количество разрядов в десятичном числе, на которое собираются спорить преподаватель и студент Коля.

Выходные данные

Если Коле имеет смысл ввязываться в этот спор - выведите «Agreed!», иначе — «No way!».

Примеры

Входные данные	Результат
5	No way!
24032018	Agreed!



Ф. Бизнес-план

Ограничение по времени: для C и C++ — 1 секунда, для Java — 2 секунды
Ограничение по памяти: 256МБ

Может Вася Пупкин и не ходит на пары, но делать домашние задания всё равно нужно. И, не смотря на то, что Вася учится на кафедре СПО (Свободного Программного Обеспечения), гуманитарных предметов там всё равно хватает. Вот, например, по Экономике Разработки Улучшений, Направленных На Дополнительную Автоматизацию (ЭРУННДА), задали написать бизнес-план для собственной фирмы, занимающейся разработкой тех-самых улучшений. «Ну что за ерунда! — думает Вася. — Почему я, будущий RobotOS разработчик, должен вместо того, чтобы изучать новый язык программирования Catlin, рассчитывать стоимость какой-то вывески на офисе какой-то фирмы, которую я никогда не открою?!». Да, бизнес-план должен быть очень подробным. Мало того, что нужно все затраты на офис рассчитать, нужно учесть даже затраты на вывеску с названием фирмы.

Вася уже нашел компанию, которая занимается изготовлением вывесок. Но заказать вывеску со своим текстом стоит очень дорого. Гораздо дешевле — заказать несколько типовых фрагментов из нескольких букв и собрать вывеску из них. Для тех, у кого не получается собрать своё название из типовых фрагментов, компания предоставляет услугу по кастомизации типовых фрагментов. За отдельную фиксированную плату можно заменить любую букву в типовом фрагменте на любую другую. Можно заменить и несколько букв, но тогда плата возрастает в соответствующее количество раз. Если купить несколько экземпляров одного и того же типового фрагмента, то в каждом экземпляре можно заменить разные буквы. А вот добавлять или убирать буквы нельзя.

Вася уже придумал название для своей фирмы. Теперь ему нужно рассчитать минимальную стоимость, за которую можно изготовить вывеску. Помогите Васе сделать это, чтобы он мог побыстрее взяться за изучение нового языка программирования Catlin.

Входные данные

В первой строке дано название фирмы (от 1 до 10^4 маленьких английских букв). Во второй строке через пробел даны два числа: N ($N \geq 1$) и p_c ($1 \leq p_c \leq 1000$, $p_c \in \mathbb{Z}$) — количество доступных типовых фрагментов и стоимость кастомизации одной буквы.

Далее идут N строк, описывающих типовые фрагменты. Каждая строка содержит текст фрагмента f_i (один или больше символов — маленьких английских букв) и его стоимость p_i ($1 \leq p_i \leq 1000$, $p_i \in \mathbb{Z}$), разделённые пробелом. Суммарная длина текстов всех фрагментов не превышает 1000 ($\text{len}(f_1) + \text{len}(f_2) + \dots + \text{len}(f_N) \leq 1000$). Тексты всех фрагментов уникальны (нет двух фрагментов с одинаковым текстом).

Выходные данные

Выведите одно целое число — минимальную возможную стоимость изготовления вывески с названием фирмы. Если же собрать вывеску из доступных типовых фрагментов невозможно ни за какие деньги, выведите число -1 .



Примеры

Входные данные	Результат
robocatlin 6 10 robot 50 robocop 80 cat 25 linux 10 atlas 40 min 15	120
algorithm 3 5 arm 35 at 20 go 15	110
helloworld 3 1 cat 8 dog 6 coders 11	-1

Пояснение

В первом примере возможны, например, такие варианты:

- купить robot (50) и заменить последнюю букву на с (10), купить atlas (40) и заменить последние две буквы на in (20)
- купить robocop (80) и заменить последние две буквы на at (20), купить min (15) и заменить первую букву на l (10) (этот вариант не оптимален)
- купить robot (50) и заменить последнюю букву на с (10), купить linux (10) и заменить все буквы на atlin (50)
- купить linux (10) и заменить все буквы на roboc (50), купить ещё один linux (10) и заменить все буквы на atlin (50)

Во втором примере оптимальный вариант — купить at (20) и заменить вторую букву на l (5), купить go (15), купить go (15) и заменить обе буквы на ri (10), купить arm (35) и заменить первые две буквы на th (10).

Г. Кролик Тэд и дерево

Ограничение по времени: для C и C++ — 1 секунда, для Java — 4 секунды
Ограничение по памяти: 256МБ

А вы знали, что кролики умеют далеко прыгать? А еще, оказывается, они умеют программировать. Более того, проводятся соревнования по программированию среди кроликов. На одном из таких команда из двух кроликов столкнулась со следующей задачей.

Дано дерево из N вершин. В дереве $N-1$ ребро и из каждой вершины можно добраться до любой другой по однозначному пути. Также в дереве есть корень, который находится в вершине с номером 1. Каждая вершина содержит грядку, в которой растёт $carrot_i$ морковок. Количество морковок может быть и отрицательным, что, как ни странно, вполне реально в кроличьем мире.

Кролику Тэду необходимо выбрать число K — длину прыжка по дереву. Затем Тэд может сколько угодно прыгать по дереву с длиной прыжка K (как в одну, так и в обратную сторону), собирая по пути морковку. Причем кролик Тэд обязательно побывает **во всех** вершинах, до которых сможет допрыгнуть, и соберет в них морковку, даже если её там отрицательное количество. Длина прыжка определяется количеством ребер на пути между двумя вершинами. Кролик не может собирать морковку с грядок в одной и той же вершине несколько раз. Заметьте, морковку с корня дерева он собирает всегда.

Необходимо найти максимальное количество морковок, которые Тэд может собрать.

Команда кроликов уже несколько часов безуспешно пытается решить эту задачу. Помогите им.



Входные данные

В первой строке содержится число N ($1 \leq N \leq 10^5$) — количество вершин в дереве. Во второй строке содержится N целых чисел $carrot_i$ ($-10^9 \leq carrot_i \leq 10^9$) — количество морковок на грядке в i -ой вершине.

В последующих $N-1$ строках содержатся пары чисел v и u ($1 \leq v, u \leq N$) — номера вершин, образующих ребра дерева.

Выходные данные

В единственной строке выведите максимальное количество морковок, которые может собрать кролик Тэд. Длина прыжка при этом должна быть $1 \leq K \leq N$.



Примеры

Входные данные	Результат
3 1 3 1 2 1 3 1	5
3 5 3 -5 3 1 2 3	8



Н. Скучная лекция

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Самое сложное в учебном процессе – занять себя во время скучных лекций. Кто-то предпочитает почитать книгу, кто-то играет в любимую игру на телефоне, кто-то мечтает о лучшей жизни. Но не Никита. Он любит вычислять в уме странные числа.

На сей раз Никита пытается найти такое число M , что у него будет ровно K делителей. Причем число M должно быть минимально возможным. Никите сложно держать в уме числа, большие за 10^9 . В связи с этим, если среди меньших чисел не нашлось такого M , то Никита считает, что ответа и вовсе не существует.

Поможете проверить ответы Никиты? Напишите программу, которая вычисляет такие числа.

Входные данные

Первая строка содержит число N ($1 \leq N \leq 320$). В следующих N строках находятся описания запросов, состоящие из одного числа K ($1 \leq K \leq 320$).

Выходные данные

Для каждого запроса выведите в отдельной строке ответ — число M , такое что M имеет ровно K делителей. При этом M должно быть минимальным возможным. Если же такого числа нет, либо оно больше 10^9 , то выведите -1 .

Примеры

Входные данные	Результат
4	1
1	2
2	16
5	-1
31	



I. Flower Beds

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

После научного семинара M математиков решили прогуляться по саду. В саду расположено N грядок. Каждая грядка состоит из некоторого множества разных цветков. Математики решили называть «правильными» такие грядки, для которых число способов составить букет из цветов грядки кратно M . В букете может быть от 1 до F_i цветков, где F_i — количество цветков в i -ой грядке. Сколько оказалось «правильных» грядок?

Входные данные

В первой строке даны два числа через пробел: M ($2 \leq M \leq 128$) — количество математиков, и N ($1 \leq N \leq 200$) — количество грядок.

Далее следует N строк. В i -ой строке — число F_i ($1 \leq F_i \leq 10^7$) — количество цветков в i -ой грядке.

Выходные данные

Выведите одно число — количество «правильных» грядок.

Примеры

Входные данные	Результат
7 2 3 1	1
2 2 2 2	0
3 3 4 5 5	1

В первом примере количество способов составить букет из цветов 1-ой грядки равно 7 и делится без остатка на количество математиков. Во втором примере из цветов каждой грядки можно составить букет тремя способами, а число 3 не кратно количеству математиков.



Ж. Смотровая точка

Ограничение по времени: для C и C++ — 1.5 секунды, для Java — 6 секунд
Ограничение по памяти: 256МБ

Вы гуляете с друзьями по городу и захотели сделать панорамное фото на 360 градусов, чтобы запечатлеть этот момент. Вокруг вас N достопримечательностей, которые вы хотите увидеть на фотографии. Каждая достопримечательность характеризуется двумя целыми числами x и y , которые задают позицию достопримечательности на карте. Никакие две достопримечательности не находятся в одной точке.



Вы хотите запечатлеть как можно больше достопримечательностей и для этого намерены выбрать наиболее выгодную позицию для фото. К сожалению, одна достопримечательность может скрывать другую, если они находятся на одном луче, исходящем из точки обзора.

Вы решили выбрать одну из имеющихся достопримечательностей как точку обзора и сделать панорамное фото из неё. Конечно же, выбирать вы будете ту, из которой будет видно как можно больше других достопримечательностей. Достопримечательность в точке обзора в панораму не попадет.

Входные данные

В первой строке находится число N ($1 \leq N \leq 10^3$). В последующих N строках содержатся координаты достопримечательностей x и y ($1 \leq x, y \leq 10^3$).

Выходные данные

В единственной строке выведите одно число — максимальное возможное количество достопримечательностей на финальном панорамном фото.

Примеры

Входные данные	Результат
3 0 0 1 0 0 1	2
4 0 0 0 1 0 2 0 3	2

К. Сложный контекст

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

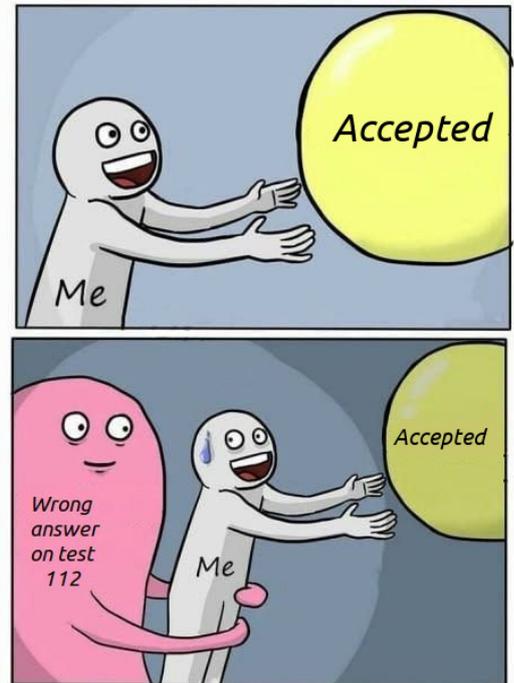
А вы когда-то ловили неправильный ответ на тесте, например, 112-ом? А вот у Жени такая проблема постоянно. Он отправляет задачу, решение которой проходит почти все тесты, но из-за глупой ошибки получает неправильный ответ на одном из последних тестов в наборе. Бывает даже такое, что Женя по итогам соревнования может не сдать одну-две такие задачи, хотя решил их идейно правильно. Такие задачи он классифицирует как «решаемые», хотя по факту не решил их (не прошел все тесты). Женя считает задачу решаемой, если решение, написанное им, прошло все тесты, либо прошло хотя-бы Q из них.

В системе тестирования для упрощения существует всего два вердикта:

1. Wrong answer on test N on problem M — неправильный ответ на тесте N задачи номер M
2. Problem M was Accepted — решение по задаче M прошло все тесты

Неправильный ответ на тесте N означает, что решение автора прошло $N-1$ тест по задаче. Во время соревнования любой участник может делать сколько угодно посылок по любой из задач.

После соревнования Жене стало интересно, сколько же задач оказались для него решенными. Он решил глянуть на логи, но их оказалось очень много. Помогите Жене написать программу, чтобы достать из логов нужную информацию.



Входные данные

В первой строке заданы числа N и Q ($1 \leq N, Q \leq 200$) — количество задач на соревновании и минимальное число тестов, чтобы считать задачу решаемой.

Во второй строке содержатся N чисел t_i ($1 \leq t_i \leq 200$) — количество тестов в i -ой задаче.

В третьей строке содержится число M ($1 \leq M \leq 200$) — количество строк в файле с логами. В последующих M строках содержатся вердикты тестирования из файла логов в хронологическом порядке, соответствуя формату вердиктов из условия.

Выходные данные

В единственной строке выведите количество задач, которые оказались решенными для Жени во время соревнования.



Примеры

Входные данные	Результат
3 5 6 22 4 3 Wrong answer on test 6 on problem 1 Problem 1 was Accepted Wrong answer on test 2 on problem 3	1
2 10 11 100 2 Wrong answer on test 10 on problem 1 Wrong answer on test 2 on problem 2	0



L. Problem1

Ограничение по времени: 1 секунда

Ограничение по памяти: 256МБ

Как и многие четверокурсники, Вася Пупкин решил в этом году устроиться на работу параллельно с учёбой. Придя на работу и познакомившись со своим коллегой Сашей, Вася принялся изучать исходный код программы, над которой ему предстоит работать. Довольно скоро у Васи возник один вопрос.

— Саша, — обращается Вася к своему коллеге, — а почему программные имена всех кнопок во всех окнах — «Button1», «Button2», «Button3»?...

— Мне так удобнее, — отвечает Саша.

— Но ведь так же легко запутаться... — возражает Вася.

— А мне — наоборот, так легче запомнить. — говорит Саша. Вот, например, во всех окнах, где есть кнопка «Сохранить», она всегда третья по счёту и называется «Button3». А, например, кнопка «Открыть» — всегда вторая и, соответственно, «Button2».

Все такие соглашения по именованию показались несовершенновыми. Поэтому он задумал всё-таки взять и по-переименовывать все кнопки. Поскольку окон и кнопок в программе очень много, делать это нужно автоматически. Но сначала, конечно же, нужно проверить, действительно ли везде соблюдаются соглашения, описанные Сашей.

Программа, над которой работают Вася и Саша, написана в среде разработки C#Builder. Графический интерфейс программы в C#Builder создаётся с помощью конструктора интерфейса и хранится в файлах форм. В каждом файле формы хранится описание интерфейса одного окна. Файл формы описывает дерево объектов — графических элементов управления, из которых состоит интерфейс. Рекомендуется посмотреть пример входных данных, прежде чем читать дальше.

Описание объекта начинается со строки вида «*object name : type*», где *name* — программное имя объекта, а *type* — тип (класс) объекта. Далее следуют несколько строк, задающих начальные значения некоторых свойств объекта в формате «*property = value*». За ними могут следовать описания вложенных объектов. Количество уровней вложенности объектов не ограничено, и объект любого типа может содержать объекты любого типа. Описание любого объекта заканчивается строкой «end».

В начале любой строки может быть несколько пробелов, но они не играют никакой роли. На концах строк пробелов не бывает. Кроме того, вокруг символов «:» и «=», а также после слова «object», всегда ровно по одному пробелу. Программные имена объектов, названия типов и свойств состоят только из букв и цифр и не бывают пустыми строками. Значения свойств могут содержать любые символы (в том числе «=»), но никогда не бывают пустыми строками.

Поскольку Вася сейчас занят своим первым заданием — разработкой встроенного просмотрщика изображений, вам придётся помочь ему с разработкой программы для проверки описанных Сашей соглашений по именованию. По заданному тексту на кнопке T , программному названию N и набору файлов форм вы должны проверить, что:

1. во всех файлах форм у всех объектов типа `TButton`, у которых начальное значение свойства `Caption` задано и равно T , программное имя равно N
2. хотя бы в одном файле формы есть хотя бы один объект, удовлетворяющий первому требованию

То есть, необходимо проверить, например, что все кнопки с текстом «Save» действительно имеют программные имена «Button3», и что где-то есть хотя бы одна такая кнопка.



Входные данные

В первой строке дана строка N (от 1 до 20 символов, английские буквы и цифры) — программное имя, которое должны иметь кнопки. Во второй строке дана строка T (от 1 до 20 символов, английские буквы, цифры и пробелы) — значение свойства `Caption` кнопок, подлежащих проверке. В третьей строке дано число F ($1 \leq F \leq 100$) — количество имеющихся в проекте файлов форм. Далее следуют описания самих файлов форм.

Описание каждого файла формы начинается с отдельной строки с числом L_i ($3 \leq L_i \leq 100$) — количеством строк в данном файле формы. За ней следует L_i строк текста (от 1 до 100 символов, маленьких и больших английских букв, цифр, пробелов и знаков : и =) — содержимое данного файла формы. Формат файла формы описан в условии. Гарантируется, что все файлы форм соответствуют формату.

Выходные данные

В единственной строке выведите результат проверки — действительно ли соблюдается проверяемое соглашение. Если оба требования из условия соблюдены, выведите Yes, иначе — выведите No.

Пример

Входные данные	Результат
<pre> Button3 Save 1 23 object Form1 : TForm1 Width = 400 Height = 200 object Label1 : TLabel Caption = File name: end object Edit1 : TEdit Width = 380 end object Panel1 : TPanel object Button1 : Tbutton Default = True Caption = Create end object Button2 : TButton Caption = Open end object Button3 : TButton Enabled = False Caption = Save end end end end </pre>	Yes

Вы можете скачать этот и ещё два примера по ссылке:

<https://acm.opu.ua/statements/onpuipc2018-for-L.txt>