

За правильное решение каждой задачи даётся 50 баллов, за неправильное (даже частичное) — 0 баллов.

По каждой задаче можно сделать не более трёх попыток.

Решения проверяются сразу после отсылки; результат сообщается участнику.

Задача А. TTS Rights

Имя входного файла:	tts.in
Имя выходного файла:	tts.out
Ограничение по времени:	5 seconds
Ограничение по памяти:	64 Megabytes

TTS (Trivial Testing System) используется в СПбГУ для того, чтобы студенты могли решать контесты из дома. TTS использует следующую систему прав. Каждый объект (команда, задача, контест, условие, и т.д.) должен состоять ровно в одной группе. Группы организованы во взвешенный ориентированный граф с целыми весами рёбер. Права объекта a на объект b — это права группы, в которой состоит a , на группу, в которой состоит b . Права группы A на группу B могут быть получены следующим способом. Система рассматривает все непустые пути из A в B . Каждому пути соответствует число, которое можно вычислить как логическое И весов всех рёбер пути. Затем система вычисляет число R — логическое ИЛИ чисел, соответствующих путям из A в B . Это число определяет права A на B . Четыре младших бита — это права приоритета 0. Они располагаются так: бит 0 это X (права на запуск), бит 1 — W (права на запись), бит 2 — R (права на чтение), бит 3 — A (права на администрирование). Тем же образом биты 4–7 определяют права приоритета 1, 8–11 — приоритета 2, и так далее. Максимальный приоритет — седьмой. Окончательные же права определяются так: система делает цикл по всем приоритетам от 0-го до 7-го. При этом для чётных приоритетов система устанавливает включённые биты, а для нечётных — обнуляет включённые биты. Полученные 4 бита и есть окончательные права.

В графе есть две вершины, `initial` и `final`. Из вершины `initial` идут рёбра веса -1 во все остальные вершины. В вершину `final` входят рёбра веса -1 из всех остальных вершин. Невозможно поменять рёбра из `initial` и в `final`.

Ваша задача — написать программу, которая поддерживает эффективное вычисление прав.

Формат входного файла

Все данные в TTS содержатся в так называемом *логе*. Лог это файл из тексто-

вых строк, каждая из которых содержит 4 или больше полей, разделённых символами табуляции (код ASCII 9). Первое поле — это всегда UNIX-время создания соответствующей записи. Второе поле определяет собственно запись. В третьем содержится описание компьютера, с которого сделана запись. Четвёртое — это идентификатор записи. Остальные поля, если они есть — это параметры записи.

В этой задаче лог будет содержать только записи о присвоении прав и о вычислении прав. Синтаксис записей следующий:

```
<utime> (<+|-|=)<GR> <ip> <src-node> <rights-mask> <{<dest-nodes>}>
<utime> (<+|-|=)<RG> <ip> <dest-node> <rights-mask> <{<src-nodes>}>
<utime> RIGHTS <ip> <src-node> <dest-node>
```

Эти команды действуют на все рёбра из `<src-node>` в `<dest-nodes>` или на все рёбра из `<src-nodes>` в `<dest-node>`. Операции с префиксом `'+'` добавляют указанные права к ребру (то есть выполняет операцию логического ИЛИ), с префиксом `'-'` убирают права (то есть обнуляют все биты, которые содержит `<rights-mask>`), а с префиксом `'='` заменяют числа на соответствующих рёбрах.

Если маска указана как `'-'`, то соответствующее число считается равным -1 .

`<rights-mask>` — это десятичное или шестнадцатеричное число. В последнем случае число начинается с префикса `0x`. Числа рассматриваются по модулю 2^{32} и не могут по абсолютной величине превосходить $2^{32} - 1$. Таким образом, каждое число, отличное от 0, имеет два различных представления, отличающихся на 2^{32} . Например, $-1 = 0xffffffff = 4294967295$. Знак минус может находиться только перед числом, например `-0x25`.

Формат выходного файла

Для каждой команды RIGHTS из входного файла, выведите сообщения о правах `<src-node>` на `<dest-node>`. Следуйте формату вывода в примере. Права должны выводиться от старших битов к младшим. Размер лога будет не больше 64 килобайт.

Примеры

tts.in									
1096000000	=GR	195.19.228.2	admin	0xffffffff	initial				
1096000000	RIGHTS	195.19.228.2	admin	initial					
1096000000	=GR	195.19.228.2	teams	-	users				
1096000000	RIGHTS	195.19.228.2	teams	users					
1096000000	+GR	195.19.228.2	users	-	nobody				
1096000000	RIGHTS	195.19.228.2	nobody	teams					
1096000000	RIGHTS	195.19.228.2	admin	nobody					
1096000000	+RG	195.19.228.2	univ	-	univ1	univ2	univ3		
1096000000	=RG	195.19.228.2	auniv	-	univ	iuniv			
1096000000	+GR	195.19.228.2	auniv	-	teams				
1096000000	+GR	195.19.228.2	contests	-	ucontests		pcontests		
1096000000	+GR	195.19.228.2	ucontests	-	trains	olymps			
1096000000	=GR	195.19.228.2	jury	0xefff	jmonitors				
1096000000	=GR	195.19.228.2	final	0x40	jmonitors				
1096000000	=GR	195.19.228.2	univ	5	ucontests				
1096000000	RIGHTS	195.19.228.2	univ1	trains					
1096000000	=GR	195.19.228.2	iuniv	5	trains				
1096000000	=GR	195.19.228.2	auniv	5	uproblems				
1096000000	=GR	195.19.228.2	nobody	5	pproblems		public		
1096000000	=GR	195.19.228.2	teams	4	statements				
1096000000	RIGHTS	195.19.228.2	univ3	statements					
1096000000	=GR	195.19.228.2	final	0x70	hidden				
tts.out									
admin has ARW rights on initial. teams has no rights on users. nobody has no rights on teams. admin has ARW rights on nobody. univ1 has RX rights on trains. univ3 has R rights on statements.									
tts.in									
1096000000	=GR	195.19.228.2	admin	0xffffffff	initial				
1096000000	=GR	195.19.228.2	admin	0xAFFFFFFF	disableadmin				
1096000000	RIGHTS	195.19.228.2	admin	disableadmin					
1096000000	=GR	195.19.228.2	kittens	-	animals				
1096000000	=RG	195.19.228.2	animals	-	dogs				
1096000000	=RG	195.19.228.2	kittens	4	animals				
1096000000	=GR	195.19.228.2	animals	4	dogs				
1096000000	=GR	195.19.228.2	animals	4	mice				
1096000000	=GR	195.19.228.2	kittens	0x3000000	mice				
1096000000	RIGHTS	195.19.228.2	kittens	mice					
1096000000	=GR	195.19.228.2	dogs	0x40	mice				
1096000000	RIGHTS	195.19.228.2	dogs	mice					
1096000000	=GR	195.19.228.2	animals	0x3f	kittens				
1096000000	RIGHTS	195.19.228.2	animals	kittens					
tts.out									
admin has RX rights on disableadmin. kittens has RWX rights on mice. dogs has no rights on mice. animals has no rights on kittens.									

Задача В. Стресс-тестирование

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

Витю пригласили в жюри межпланетной олимпиады по программированию. Это большая честь для него, и он постарался не ударить в грязь лицом. Он реализовал технологию, которая позволяет находить интересные тесты для задач по имеющимся решениям жюри. Эта технология называется *стресс-тестирование*.

Поиск теста осуществляется по следующей схеме. Сначала пишется генератор общего случая теста к задаче, т.е. такой генератор, который с достаточной вероятностью выдает тесты на все случаи, которые необходимо разобрать. Он не обязательно является генератором полностью случайного теста, т.к. зачастую на таких тестах реализуются далеко не все случаи решения.

Генератор, естественно, использует случайные числа для создания теста. Он инициализируется с помощью функции randomize, которая берет случайное значение для инициализации генератора случайных чисел из внешнего источника (как правило, в качестве такого источника используется системное время). Для того чтобы можно было потом восстановить тест, значение, которым был проинициализирован генератор случайных чисел, выдается в файл протокола. Всегда гарантируется, что при одной и той же инициализации генератор случайных чисел будет выдавать одну и ту же последовательность.

После генерации теста на нем запускаются два различных решения жюри. Специальная программа gup замеряет время работы каждого из решений. Естественно, вся информация о времени работы, параметрах запуска и используемой памяти также выводится в файл протокола работы системы.

Затем происходит сравнение выходных файлов обеих программ. К сожалению, в связи с особенностями используемой суперсовременной операционной системы Doors, информацию о сравнении не так-то просто вывести в файл протокола, поэтому Витя ограничился лишь прерыванием тестирования в случае, если одно из решений выдало неверный ответ.

Весь этот процесс запускается в цикле, пока не будет найден тест, на котором одна из программ выдает неверный ответ, или же член жюри, занимающийся тестированием, не сочтет, что уже было проверено достаточное количество различных тестов.

В первом случае, очевидно, интересным является тест, на котором завалилось решение жюри. Во втором же случае Витя считает интересными тесты, на которых

каждое из двух решений работало максимальное время. К сожалению, исходный файл протокола не очень удобен для получения этой информации.

Необходимо написать программу, которая бы преобразовывала файл протокола в более удобный формат, а также находила бы тесты, на которых каждое из решений работало максимальное время.

Формат входного файла

Во входном файле содержится исходный файл протокола. Его размер не превосходит двух мегабайт. Он состоит из одного или нескольких блоков, каждый из которых соответствует запуску обоих решений на одном тесте. Каждый из этих блоков начинается и заканчивается непустой строкой, состоящей из одних символов ‘-’.

Внутри блока содержится информация об инициализации генератора случайных чисел, а также о времени работы каждого из решений. Инициализация генератора задается строкой в формате **randseed** = x , где x — целое неотрицательное число, не превосходящее 10^9 . За ней следуют протоколы запуска решений. Время работы каждого из решений выведено в формате **Work time**: x_i ms, где x_i — вещественное число, в котором в качестве разделителя целой и дробной части используется запятая — так уж вывела операционная система Doogs. Целая часть этого числа не превосходит 10^6 . Витя знает, что операционная система Doogs все равно производит измерения таким образом, что дробные части совпадают при совпадении целых частей, поэтому его интересует только целая часть этого числа.

Кроме указанных строк, в файле протокола может содержаться также другая информация, которую Витя пока не хочет анализировать.

Всегда сначала запускается первое решение жюри, а затем второе. Кроме того, запуск решений происходит только после генерации теста, а, следовательно, и вывода в файл протокола информации об инициализации генератора случайных чисел.

Формат выходного файла

Для каждого из блоков в порядке их следования должны быть выведены три строки информации. В первой строке выводится значение, которым был проинициализирован генератор случайных чисел, в формате **At randseed** = x . Во второй и третьей строках выводится время работы первого и второго решений соответственно, в виде **First**: x_1 ms и **Second**: x_2 ms, где x_i — время работы каждого из решений с отброшенной дробной частью.

Кроме этого, необходимо собрать информацию о том, на каких тестах каждое из решений работало максимальное время. Эта информация выводится после всей информации о блоках. Если какое-то решение работало одинаково долго на

нескольких тестах, то выбирается самый первый из блоков, на котором было достигнуто такое время.

Проверка производится автоматически, поэтому вывод вашей программы должен по возможности максимально совпадать с эталонным выводом!

Пример

stress.in
----- randseed = 43794135 Testing ..\b_mb, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (80 ms) Work time: 80,11520 ms Maximal memory usage: 10336 Kb Testing ..\b_al, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (391 ms) Work time: 390,56160 ms Maximal memory usage: 2680 Kb ----- randseed = 43903502 Testing ..\b_mb, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (110 ms) Work time: 110,15840 ms Maximal memory usage: 10328 Kb Testing ..\b_al, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (310 ms) Work time: 310,44640 ms Maximal memory usage: 2680 Kb -----
stress.out
At randseed = 43794135 First: 80 ms Second: 390 ms At randseed = 43903502 First: 110 ms Second: 310 ms Maximal work time for first: 110 at randseed = 43903502 Maximal work time for second: 390 at randseed = 43794135

Задача С. Логи Apache

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

21 декабря в Центр была передана информация от Штирлица — лог сервера

Apache, отражающий доступ к ключевому интернет-ресурсу, расположенному в государстве Borderland. В результате успешной спецоперации в Borderland сложилась тупиковая ситуация вокруг референдума о том, какая из букв нового алфавита должна быть первой. После предыдущих итераций выбор сузился до Y и Z. Очередная итерация процесса выхода из кризиса намечалась уже через пять дней. Для того чтобы восстановить последовательность запросов, сотрудники Центра, не совсем пришедшие в себя после Дня Чекиста, решили перевести события лога к временной зоне секретной базы, на которой разрабатывается новая диверсия.

Формат входного файла

В первой строке записана временная зона секретной базы, к которой нужно преобразовать лог. Начиная со второй строки, задан сам лог. Строки лога Apache всегда имеют вид

$\langle ip \rangle - \langle username \rangle \langle date \rangle \langle extra\ fields \rangle$

Поле *ip* представляет собой ip-адрес узла, сделавшего запрос, в стандартном формате. Второе поле всегда содержит один символ ‘-’. Имя пользователя состоит из произвольного ненулевого количества латинских букв, символов ‘_’ и кавычек. Формат даты следующий:

$$[\langle \text{день} \rangle / \langle \text{месяц} \rangle / \langle \text{год} \rangle : \langle \text{часы} \rangle : \langle \text{минуты} \rangle : \langle \text{секунды} \rangle \langle \text{временная зона} \rangle]$$

Временная зона всегда записывается ровно пятью символами:

$$\langle \text{знак} \rangle \langle \text{часы} \rangle \langle \text{минуты} \rangle$$

Знак принимает значение ‘+’, если часы в данной временной зоне опережают часы в зоне GMT (Greenwich Mean Time), в противном случае ставится знак ‘-’.

День, часы, минуты и секунды всегда состоят ровно из двух цифр (как и в поле даты, так и во всех записях временных зон), год — из четырех. Месяц записан сокращением на английском языке и может принимать одно из следующих значений: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

Допустимые во входном файле номера года лежат в диапазоне от 1980 до 2099. Високосными считаются года, номер которых делится на 4.

Дополнительные поля состоят из произвольного количества символов с кодами от 32 до 126.

В записях временной зоны поле часов всегда содержит значение, не превосходящее 12. Размер входного файла не превосходит одного мегабайта. Все даты корректны и проверка их не требуется.

Формат выходного файла

Результатом работы программы является лог Apache, в котором все времена и даты преобразованы к заданной временной зоне. Все остальные поля в каждой строке лога должны оставаться без изменений. Имейте в виду, что проверка производится автоматически, поэтому все даты и времена должны содержать по две цифры в полях дней, часов, минут и секунд, три символа в поле месяца (case-sensitive), и четыре цифры в поле года.

Пример

apache.in
-0200 195.19.224.104 - abc [02/Dec/2004:18:25:19 +0100] "GET / HTTP/1.1" 304 - 195.19.224.104 - - [02/Dec/2004:18:25:19 +0100] "GET /main.css HTTP/1.1" 304 - 195.19.224.79 - - [02/Dec/2004:20:58:20 +0300] "GET /tts HTTP/1.1" 302 293 195.19.224.79 - - [02/Dec/2004:20:58:20 +0300] "GET /tts/ HTTP/1.1" 302 303 207.46.98.42 - - [03/Dec/2004:04:39:32 +0300] "GET /robots.txt HTTP/1.0"
apache.out
195.19.224.104 - abc [02/Dec/2004:15:25:19 -0200] "GET / HTTP/1.1" 304 - 195.19.224.104 - - [02/Dec/2004:15:25:19 -0200] "GET /main.css HTTP/1.1" 304 - 195.19.224.79 - - [02/Dec/2004:15:58:20 -0200] "GET /tts HTTP/1.1" 302 293 195.19.224.79 - - [02/Dec/2004:15:58:20 -0200] "GET /tts/ HTTP/1.1" 302 303 207.46.98.42 - - [02/Dec/2004:23:39:32 -0200] "GET /robots.txt HTTP/1.0"

Задача D. Статистика звонков

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

Андрюша постоянно жалуется папе, что его младший братик Саша часами сидит в интернете и не даёт ему готовиться к олимпиаде. Папа решил всерьёз заняться ситуацией, т.к. Саша действительно, похоже, превышает отведённый ему на интернет лимит времени. Для выяснения того, сколько же Саша просидел в интернете, папа решил написать анализатор логов модема от Windows’95. Единственная проблема заключается в том, что Windows ведет лог в крайне неудобочитаемом формате и выводит туда слишком много бессмысленной информации...

Лог модема Windows состоит из строчек, начинающихся датой и временем в формате mm-dd-yyuu hh:mm:ss.ss, т.е. две цифры месяца, дефис, две цифры дня, дефис, четыре цифры года, пробел, часы, двоеточие, минуты, двоеточие, секунды, точка, сотые доли секунд. Затем следует пробел, дефис, еще один пробел и далее сама строчка.

Например:

