

Instruções gerais: O BOCA é um sistema de correção automática de exercícios que verifica se o resultado gerado pelo seu programa satisfaz casos de teste pré-definidos. Portanto, é necessário seguir estritamente os formatos especificados na questão. Lembre-se que os exemplos dados servem para facilitar o entendimento e podem não cobrir todos os casos de teste que serão usados.

Para alguns *schedules*, é fácil recuperar-se de falhas de transação e sistema, enquanto para outros o processo de recuperação pode ser bem complicado. Em alguns casos, nem sequer é possível recuperar-se corretamente após uma falha. Portanto, é importante caracterizar os tipos de *schedules* para os quais a *recuperação é possível*, bem como aqueles para os quais a *recuperação é relativamente simples*. Essas caracterizações não oferecem de fato o algoritmo de recuperação; elas só tentam caracterizar de modo teórico os diferentes tipos de *schedules*.

Um *schedule S* é **recuperável** se nenhuma transação *T* em *S* for confirmada até que todas as transações *T'*, que tiverem gravado algum item *X* que *T* lê, sejam confirmadas. Uma transação *T* lê da transação *T'* em um *schedule S* se algum item *X* for gravado primeiro por *T'* e depois lido por *T*. Além disso, *T'* não deve ser cancelada antes que *T* leia o item *X*, e não deve haver transações que gravam *X* depois que *T'* o grave e antes que *T* o leia (a menos que essas transações, se houver, forem abortadas antes que *T* leia *X*).

Escreva uma consulta que verifica se um *schedule* é ou não recuperável.

Entrada:

Considere a existência da tabela **Schedule**, na qual cada linha representa a chegada de uma operação pertencente a uma dada transação (o número de transações presentes no *schedule* pode variar). A tabela possui 4 colunas: a primeira representa o tempo de chegada (time), a segunda o identificador da transação (#t), a terceira a operação (read_lock : bloqueio compartilhado para leitura de um item, write_lock : bloqueio (exclusivo) para escrita/gravação de um item, unlock : desbloqueio de um item, read_item : leitura de um item, write_item : escrita de um item, commit : confirmação ou rollback : aborto/rollback) e a quarta o item de dados (atributo) que será bloqueado/desbloqueado/lido/escrito (quando aplicável). As linhas da tabela estão ordenadas logicamente pelo valor na primeira coluna, que indica o carimbo (rótulo) de tempo (*timestamp*) de chegada (quanto menor o valor, mais antiga a operação).

Saída:

A saída deve ser uma tabela contendo uma coluna chamada RESP com o valor 1, se o *schedule* for recuperável; caso contrário, 0.

Exemplo 01

time	#t	op	attr
1	1	read_item	X
2	2	read_item	X
3	1	write_item	X
4	1	read_item	Y
5	2	write_item	X
6	2	commit	-
7	1	write_item	Y
8	1	commit	-

Saída

1

Exemplo 02

time	#t	op	attr
15	1	read_item	X
17	1	write_item	X
22	2	read_item	X
23	1	read_item	Y
26	2	write_item	X
34	1	write_item	Y
35	1	commit	-
37	2	commit	-

Saída

1

Exemplo 03

time	#t	op	attr
41	1	read_item	X
72	1	write_item	X
83	2	read_item	X
84	1	read_item	Y
85	2	write_item	X
96	2	commit	-
117	1	write_item	Y
128	1	rollback	-

Saída

0

Exemplo 04

time	#t	op	attr
141	2	read_item	X
152	2	write_item	X
153	1	read_item	X
154	1	write_item	X
165	2	rollback	-
186	1	read_item	Y
197	1	write_item	Y
208	1	commit	-

Saída

0