

G. Arquivo Corrompido

time limit per test: 1 second

memory limit per test: 256 megabytes

Agostinho começou a trabalhar no IPD (Instituto de Processamento de Dados) e na primeira semana de trabalho, recebeu um dos maiores problemas da sua carreira: o banco de dados que mantém o registro dos alunos da faculdade naquele semestre, nas disciplinas que eles estão cursando, foi corrompido.

Ele sabe que um aluno pode cursar diversas disciplinas em um semestre, mas um mesmo aluno não pode cursar a mesma disciplina mais de uma vez no mesmo semestre. Assim, ele foi requisitado a verificar se isso ocorre em algum arquivo, sendo um sinal de que o arquivo está corrompido.

Como Agostinho sabe que você é um bom programador, ele pediu a sua ajuda para escrever um programa que, dadas essas condições, verifique se um arquivo está corrompido ou não.

Input

A entrada possui diversos casos de testes, cada um representando um arquivo diferente. A primeira linha da entrada contém um inteiro T ($1 \leq T \leq 25$) que indica o número de arquivos a serem verificados. Cada caso de teste começa por uma linha que contém um inteiro L ($1 \leq L \leq 10000$) que são a quantidade de linhas do arquivo a ser verificado. Em seguida virão L linhas, onde cada linha L_i contém duas informações separadas por um espaço, um inteiro de seis dígitos, que representa a matrícula do aluno, e uma string composta por letras maiúsculas, minúsculas ou dígitos, que representam o nome da disciplina que o aluno "pode" estar cursando.


Output

Para cada caso de teste, deve-se imprimir uma linha contendo "Arquivo OK" se o arquivo não apresentar nenhum problema, ou "Corrompido com X erro(s)" onde X é o número de inconsistências encontradas no arquivo.

Examples

input
<pre>1 6 123456 algoritmos 123456 redes 123456 empreendedorismo 654111 redes 654111 algoritmos 333444 empreendedorismo</pre>
output
Arquivo OK

input
<pre>2 3 112233 alp 112266 c1 112233 alp 4 357159 c2 357159 ia 222222 taa 357159 taa</pre>
output
<pre>Corrompido com 1 erro(s) Arquivo OK</pre>

IDP - TAA - 2025/02
Private
Participant


→ **About Group**



Este grupo tem o objetivo de organizar as atividades de programação da disciplina de Técnicas de Programação e Análise de Algoritmos.

[Group website](#)

→ **Group Contests**

- TAA - LEA 05
- TAA - LEE 05
- TAA - LEA 04
- TAA - LEE 04
- TAA - AS 01
- TAA - LEA 03
- TAA - LEE 03
- TAA - LEA 02
- TAA - LEE 02
- TAA - LEA 01
- TAA - LEE 01
- ET - Exercício de Testes

TAA - LEE 04
Finished
Contestant


→ **Last submissions**

Submission	Time	Verdict
344771152	Oct/19/2025 22:42	Accepted

Supported by

