MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema D

Injeção de dependências

Arquivo fonte: injecaodependencia.{ c | cpp | java | py }
Autor: Prof. Dr. Leandro Luque (Fatec Mogi das Cruzes)

A gestão eficiente de dependências é crucial para reduzir o acoplamento em sistemas de software, promovendo a modularidade e a reusabilidade do código. Dentre as técnicas para gerenciar dependências, a Injeção de Dependência e o padrão *Service Locator* são amplamente reconhecidos. Na Injeção de Dependência, uma classe não instancia diretamente os objetos das classes das quais depende; ao invés disso, ela os recebe através do construtor, de atributos ou de métodos de acesso.

Desafios surgem quando existem dependências cíclicas, como no caso de A -> B, B -> C e C -> A. Para instanciar A, é necessário instanciar e injetar B, que por sua vez depende de C, que depende de A, formando um ciclo. Uma solução é empregar a "injeção tardia", onde A pode ser criado sem a imediata injeção de B, que ocorrerá após a criação de todas as dependências.

Munarinho está elaborando um *framework* de injeção de dependências para um novo projeto de software e precisa que este identifique automaticamente dependências cíclicas entre os módulos do projeto.

Entrada

A primeira linha da entrada contém um inteiro n ($1 \leq n \leq 500$), representando o número de dependências no projeto. As próximas n linhas contêm dois caracteres maiúsculos a e b indicando que o módulo a depende do módulo b.

Saída

A saída deve apresentar a string "usar injecao tardia"se for identificada uma dependência cíclica no projeto; caso contrário, deve apresentar "ok".

Exemplo de Entrada 1	Exemplo de Saída 1
3 A B B C C A	usar injecao tardia

Exemplo de Entrada 2	Exemplo de Saída 2
2 A B C D	ok