

Gloud Computing

Por Lucas Hermann Negri, UDESC  Brasil

Timelimit: 1

A Gloud Computing está vindo se instalar para a região de Joinville. Eles são conhecidos por proverem aplicativos na internet, mais especificamente um modelo de negócios baseado em cloud computing - computação nas nuvens.

A fim de selecionar os novos funcionários da empresa, eles contactaram o comitê da maratona da UDESC, para que passassem um problema aos nossos maratonistas. Aquele que resolver, além do balão, pode preencher a ficha funcional com estrelinhas a mais.

Basicamente, a Gloud Computing tem aplicações espalhadas em seus servidores em diversos lugares do mundo. Estes servidores são especializados em uma lista de aplicativos a serem usados pelos usuários ali conectados na internet das nuvens.

Por exemplo, o servidor de Joinville pode disponibilizar a aplicação **A**, enquanto que o de Pasadena na Califórnia provê as aplicações **A**, **B** e **C** e o servidor de Pomerode provê a aplicação **C**.

Temos um conjunto de servidores a cada um com um conjunto de aplicações a serem disponibilizados a um conjunto de usuários. Cada usuário pode estar conectado a um ou mais servidores dependendo de sua demanda, como ilustrado na Figura 1.

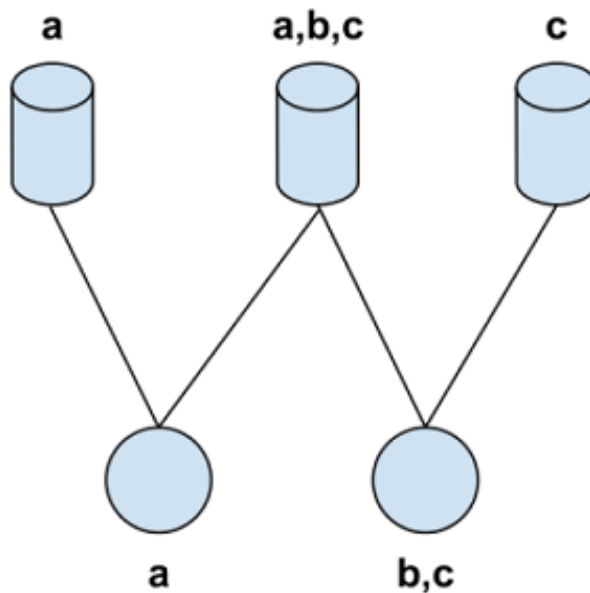


Figura 1: 3 provedores de serviços, 2 usuários e 4 conexões.

Serão disponibilizados a você dados sobre estes dois conjuntos, servidores e demanda dos usuários, e você deverá dizer a quantidade total de conexões entre clientes e servidores. As conexões são feitas de forma a maximizar a redundância. Por exemplo, se um cliente utilizar as aplicações **B** e **C**, ele irá se conectar a todos os servidores que disponibilizarem ao menos a aplicação **B** e a todos os que disponibilizarem ao menos a **C**. Múltiplas conexões entre um mesmo par de cliente e servidor são contabilizadas como um só. Pode ser que um cliente requeira uma aplicação inexistente, assim como o caso de um servidor prover uma aplicação não requisitada por nenhum cliente.

Entrada

A entrada é composta por vários casos de teste. Cada caso de teste é iniciado por dois inteiros, **N** e **M** ($0 \leq N, M \leq 200$), que correspondem ao número de servidores e ao número de clientes. Cada uma das próximas **N** linhas contém um valor **Qi** ($0 \leq Qi \leq 100$) correspondente ao número de aplicações fornecidas pelo i-ésimo servidor, seguido por **Qi** palavras (separadas por espaços) referentes aos nomes das aplicações fornecidas. Após esta descrição dos servidores, seguem **M** linhas, cada uma contendo um valor **Pj** ($0 \leq Pj \leq 100$) correspondente ao número de aplicações requisitadas pelo j-ésimo cliente, seguido por **Pj** palavras (separadas por espaços) referentes aos nomes das aplicações requisitadas. A entrada termina quando **N** = **M** = 0. Todos os nomes de aplicativos tem tamanho entre 1 e 20 caracteres.

Saída

Para cada caso de teste, o programa deve imprimir a soma total de conexões entre cliente e servidores em uma linha, desconsiderando múltiplas conexões entre um mesmo par de cliente e servidor.

Exemplo de Entrada	Exemplo de Saída
3 2 1 a 3 a b c 1 c 1 a 2 b c 5 2 2 s1 s2 2 s3 s4 2 s5 s6 2 s7 s8 2 s1 s2 3 s1 s2 s20 3 s1 s2 s21 0 0	4 4