



Introdução à Ciência da Computação - 113913

Prova 2

Questão B

Observações:

- As provas também serão corrigidas por um **corretor automático**, portanto é necessário que as entradas e saídas do seu programa estejam conforme o padrão especificado em cada questão (exemplo de entrada e saída). Por exemplo, não use mensagens escritas durante o desenvolvimento do seu código como “Informe a primeira entrada”. Estas mensagens não são tratadas pelo corretor, portanto a correção irá resultar em resposta errada, mesmo que seu código esteja correto.
- Serão testadas várias entradas além das que foram dadas como exemplo, assim como as listas.
- Assim como as listas, as provas devem ser feitas na versão Python 3 ou superior.
- **Questão A valerá 40% da nota da Prova 2 e a Questão B valerá 60% da nota da Prova 2.**
- Leia com atenção e faça **exatamente** o que está sendo pedido.

Questão B - Game Design

Um conceito muito conhecido no game design, na animação e na criação de robôs é o *Uncanny Valley*. Ele é a hipótese de que réplicas humanas que parecem quase humanas, porém não completamente, despertam um sentimento de repulsa e desconforto em seres humanos.

Depois do sucesso do seu jogo Elastiman, Roberto mais uma vez veio ao seu contato. Ele está desenvolvendo um jogo realista, mas está se encontrando preso no *Uncanny Valley* por alguns diversos motivos. Seu trabalho será reconhecer quais as *features* que estão causando a perda de “humanidade” dos seus modelos 3D.

Entrada

A primeira linha da entrada contém um inteiro **N**, o número de features a serem descritas. As próximas **N** linhas contêm, cada uma, uma string **F_i**, o nome da *feature* descrita. A linha seguinte, por sua vez, contém um inteiro **M**, o número de modelos a serem analisados.

As próximas **M** linhas contêm, cada uma, por fim, uma string de **N** caracteres, cada um podendo ser **X**, caso o modelo atenda à *feature*, ou **O** (a letra ‘O’), caso contrário. Ou seja, se o **primeiro** caractere da string for um ‘X’, esse modelo atende a **primeira** *feature*.

Saída

Seu programa deve identificar dentre os modelos para análise, qual o que menos atende às *features* definidas (ou seja, o que mais contém ‘O’s) e imprimir na saída padrão quais as *features* que ele **não** atende, na ordem de input, um em cada linha.

Considere que sempre haverá um único modelo com o número mínimo de **features** atendidas.

Exemplo de Entrada	Exemplo de Saída
3 olhos vibrantes texturas sólidas cabelo com reflexão 2 OXO XOX	olhos vibrantes cabelo com reflexão
2 ciclo de caminhada natural animações de diálogo não-robóticas 5 XX OO XO OX XX	ciclo de caminhada natural animações de diálogo não-robóticas

Boa Prova!