



**Universidade de Brasília**

Departamento de Ciência da Computação

## **Introdução à Ciência da Computação - 113913**

### **Prova G**

#### **Observações:**

- Assim como as listas de exercícios, as provas serão corrigidas por um **corretor automático**, portanto é necessário que as entradas e saídas do seu programa estejam conforme o padrão especificado (exemplo de entrada e saída).
- Por este motivo, nunca use mensagens escritas para requisitar input (e.g. 'Informe o número de casos de uso'). Estas mensagens são consideradas parte do output do seu programa e resultarão em Resposta Errada, mesmo que o resto do seu código esteja correto.
- Leia com atenção e faça **exatamente** o que está sendo pedido.
- Assim como as listas, as provas devem ser feitas utilizando **Python 3**. Use esta versão do Python.

# 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.

## Exemplos

Entrada	Saída
3 olhos vibrantes texturas sólidas cabelo com reflexão 2 OXO XOX	olhos vibrantes cabelo com reflexão
Entrada	Saída
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