



## DESAFIO DE PROGRAMAÇÃO - ACADEMIA CAPGEMINI

Olá! Seja bem-vindo (a) à terceira etapa do processo de seleção para a Academia Capgemini 2022. O objetivo dessa etapa é testar os seus conhecimentos em lógica de programação. Para isso, preparamos três questões com diferentes níveis de dificuldade. *A implementação das questões pode ser feita em qualquer linguagem, porém a utilização de Java será um diferencial.*

### # Questão 01

Escreva um algoritmo que mostre na tela uma escada de tamanho **n** utilizando o caractere **\*** e espaços. A base e altura da escada devem ser iguais ao valor de **n**. A última linha não deve conter nenhum espaço.

**Exemplo:**

**Entrada:**

```
n = 6
```

**Saída:**

```
*
**
***
****
*****
*****
```

### # Questão 02

Débora se inscreveu em uma rede social para se manter em contato com seus amigos. A página de cadastro exigia o preenchimento dos campos de nome e senha, porém a senha precisa ser forte. O site considera uma senha forte quando ela satisfaz os seguintes critérios:

- Possui no mínimo 6 caracteres.
- Contém no mínimo 1 dígito.
- Contém no mínimo 1 letra em minúsculo.
- Contém no mínimo 1 letra em maiúsculo.
- Contém no mínimo 1 caractere especial. Os caracteres especiais são: `!@#$%^&*()-+`



Débora digitou uma string aleatória no campo de senha, porém ela não tem certeza se é uma senha forte. Para ajudar Débora, construa um algoritmo que informe qual é o número mínimo de caracteres que devem ser adicionados para uma string qualquer ser considerada segura.

**Exemplo:**

**Entrada:**

Ya3

**Saída:**

3

**Explicação:**

Ela pode tornar a senha segura adicionando 3 caracteres, por exemplo, &ab, transformando a senha em Ya3&ab. 2 caracteres não são suficientes visto que a senha precisa ter um tamanho mínimo de 6 caracteres.

**# Questão 03**

Duas palavras podem ser consideradas anagramas de si mesmas se as letras de uma palavra podem ser realocadas para formar a outra palavra. Dada uma string qualquer, desenvolva um algoritmo que encontre o número de pares de substrings que são anagramas.

**Exemplo:**

**Exemplo 1)**

**Entrada:**

ovo

**Saída:**

2

**Explicação:**

A lista de todos os anagramas pares são: [o, o], [ov, vo] que estão nas posições [[0, [2]], [[0, 1], [1, 2]] respectivamente.



### Exemplo 2)

#### Entrada:

```
ifailuhkqq
```

#### Saída:

```
3
```

#### Explicação:

A lista de todos os anagramas pares são: [i, i], [q, q] e [ifa, fai] que estão nas posições [[0], [3]], [[8], [9]] e [[0, 1, 2], [1, 2, 3]].

### # O que será avaliado

- Documentação
- Estrutura do código
- Atendimento aos requisitos
- Testes unitários

### # Envio das questões

As soluções para as questões devem ser hospedadas no GitHub e o link do repositório deve ser postado na sua área do candidato a partir do dia 14/02/2022. Para entrar na sua área do candidato acesse: <https://capgemini.proway.com.br/inscricao/login.php>. O link do repositório deve ser postado no campo “Github para o desafio de programação”. O link deve ser similar a este: <https://github.com/nome-de-usuario/repositorio>. Lembrando que a data final para postagem do desafio será no dia **20/02/2022**. Quanto antes você fizer, maiores as chances de ser selecionado (a) para a próxima etapa. 🚀

O repositório deve conter um arquivo README.md com as instruções de como rodar a aplicação e as tecnologias utilizadas.