

## Case do Desafio for\_code

Escrito por [Diretoria de Projetos](#) em 10/03/2025

### O que é esse documento

Este documento explica o case do desafio for\_code do processo seletivo de 2025.1, informando as questões e como será esperado a feitura das mesmas. Utilize o [Documento de Suporte - Python](#) para auxiliá-lo na feitura dos exercícios.

OBS: As questões não estão em ordem de dificuldade!



### O CASE

#### 1. Van der Waals

Você foi alocado em uma iniciação científica que utiliza a computação como ferramenta dentro da termodinâmica, um assunto muito importante dentro da engenharia química. Então, enquanto membro da for\_code, por já saber programar na linguagem python, seu orientador te aloca para a tarefa de criar um gráfico de Pressão (eixo vertical y) pelo volume molar (eixo horizontal x).

A pressão será modelada pela equação de Van der Waals a seguir, que depende de T:

$$P = \frac{RT}{\bar{V}-b} - \frac{a}{\bar{V}^2}$$

Em que a e b são parâmetros não nulos que dependem da substância cujas propriedades PVT se deseja analisar, com **Temperatura em Kelvin**, **Volume em m³/mol** e **Pressão em Pa**. Além disso, b é uma constante chamada volume de exclusão ou covolume e é chamado de parâmetro de coesão.

Ademais, a e b podem ser calculados da seguinte forma:

## for\_code[ ]

$$a = \frac{27}{64} \frac{R^2 T_c^2}{P_c}; b = \frac{1}{8} \frac{RT_c}{P_c}$$

Em que R é a constante dos gases ideais e T<sub>c</sub> e P<sub>c</sub> são, respectivamente, a temperatura e a pressão críticas da substância que podem ser encontradas em tabelas de dados ou em sites como o NIST.

Sabendo as fórmulas acima, o seu objetivo é plotar este gráfico para T = 41°F para gás carbônico puro.

### Dicas:

- Se atente ao covolume b no denominador, cuidado com o infinito!
- R = 8,314 J/(K mol)
- T<sub>c</sub> = 304,2 K
- P<sub>c</sub> = 73,83e5 Pa
- Se atente à unidade da temperatura a ser calculada

## 2. App de Música

Sua equipe está trabalhando em um app de streaming de músicas e uma das funcionalidades é criar um **embaralhador de músicas**. Uma pesquisa feita pela equipe de UX (experiência do usuário) mostrou que essa é uma das funcionalidades mais importantes para os usuários e por isso foi priorizada a criação de um experimento para testar a melhor solução.

A ideia é ter vários embaralhadores diferentes e realizar um teste com partes dos usuários (chamado de teste A/B), onde cada grupo de usuários selecionado recebe uma versão e através de pesquisas e métricas de utilização saberemos qual código terá a maior **eficiência**.

Sua tarefa será desenvolver um desses embaralhadores. Você deve criar uma **função** que receberá uma **lista de pesos** que representa as **músicas mais ouvidas** pelo usuário. Sua função deve **retornar** uma **lista** organizada intercalando as músicas **mais** ouvidas com as músicas **menos** ouvidas. Por exemplo:

# for\_code[ ]

Na situação onde a lista de pesos é: [2, 10, 5, 3] sua função deverá retornar [10, 2, 5, 3]

## Dica:

- Existe um método de listas, que deixa essa lista de entrada mais familiar para essa questão.
- Admita que a entrada sempre será uma lista de inteiros.
- Não se limitem apenas a este teste, a função deve cobrir todos os erros.

## 3. Elefante

Você está lecionando uma aula de programação e gostaria de mostrar o poder da lógica. Sendo assim, como um bom educador, lembrou de uma música específica pois sabia que nela havia um certo padrão acontecendo. Pensando nisso, desenvolveu um código que gerava a letra da música dinamicamente para que o retorno fosse compatível com a entrada `n` elefantes especificada.

Na primeira parte da música gerada:

- Um elefante, você deve escrever por extenso e no singular ("Um elefante INCOMODA...")
- Para os demais, utilize números e o plural ("n elefantes INCOMODAM...")

Na segunda parte da música gerada:

- Após a primeira parte, a palavra INCOMODAM deve ser repetida de forma dinâmica, de acordo com `n` atual de elefantes
- Ao retornar para a primeira parte, leve em consideração a quantidade de vezes que INCOMODAM repetiu na segunda parte

Exemplo:

para `n = 3`

Um elefante `incomoda` muita gente

2 elefantes `incomodam incomodam` muito mais

2 elefantes **incomodam** muita gente

3 elefantes **incomodam incomodam incomodam** muito mais

Dicas:

- Lembre-se que é possível concatenar **strings** com o operador "+"

## 4. Pedra papel e tesoura

Tendo bastante dificuldades para fazer amigos, mas ainda querendo jogar jogos simples de duas pessoas, João se apaixonou por jogar **pedra-papel-tesoura** no computador, em um site de jogos online.

Infelizmente a internet caiu em sua casa, e ele agora está sem sua diversão particular. Entretanto, ele possui **Python** instalado no seu computador, e resolveu **programar o jogo** para ele mesmo.

Agora, sua missão é programar um programa que **jogue pedra-papel-tesoura** contra o usuário, que revele **quem ganhou a cada rodada** e mantenha um **placar** das vitórias, derrotas e empates do jogador.

Dicas:

- Tente primeiro fazer um programa que joga uma única partida e determina vitória, derrota ou empate. Em seguida, tente criar um loop centrado nesse programa anterior.

## 5. Brincando com a Matrix

Em um reino distante, existia um cofre mágico protegido por um enigma matemático. Para abrir o cofre, era necessário **encontrar a coluna mais poderosa** de uma **Matriz** do Destino. Mas não era qualquer coluna! A chave do cofre estava escondida na coluna **cuja soma dos números era a maior** e, ao mesmo tempo, **sua multiplicação também era a maior** possível.

# for\_code[ ]

O sábio mago Pytheon, encarregado de proteger o cofre, lançou um desafio aos aventureiros:

"Encontrem as **colunas** de maior poder, **onde tanto a soma quanto a multiplicação alcancem seus valores máximos**. Se houver mais de uma coluna com os **mesmos valores**, todas deverão ser reveladas, pois nelas está o segredo do cofre!"

Ajude a realizar esse problema! Para isso, precisamos fazer um código que identifique essas colunas em Python.

Crie uma função cuja **entrada** deve ser uma **matriz** com números inteiros e a **saída** uma tupla com **duas listas** dentro, sendo essas listas com os **índices** dessas maiores somas e multiplicações desses números.

- Exemplo:

- Entrada:

```
[  
    [1, 2, 3, 3],  
    [4, 5, 6, 6],  
    [7, 8, 9, 9]  
]
```

- Saída esperada: [2,3] , [2,3]

Além disso, é bom avisar qual é a soma e qual multiplicação, como exemplo:

Índice da coluna com maior soma: [2,3]

Índice da coluna com maior multiplicação: [2,3]

# for\_code[ ]

## Dicas:

- Faça bom uso do `index()` e do `max()` pois esses serão essenciais.
- Inicialize com listas já prontas da soma e da multiplicação, sendo respectivamente “`soma = [0] * numero_de_colunas`” e “`mul = [1] * numero_de_colunas`”. Aí, modificá-las para ver a maior no final.

$$A = \begin{bmatrix} 3 & 1 \\ 0 & -2 \end{bmatrix}$$

$m$  LINHAS

$n$  COLUNAS



## Prazo

De acordo com a experiência da for\_code na feitura de projetos, indicamos um cronograma que pode facilitar na feitura das questões. Lembrando que esta é apenas uma sugestão, então sintam-se à vontade para fazer as questões da maneira que preferirem:

- 7 a 13 de abril:
  - Fazer questões 1 e 2.
- 14 a 22 de abril:
  - Fazer questões 3, 4 e 5.

É interessante lembrar também que ao mesmo tempo que o case, vocês tem também o Pitch a ser feito, então se organizem e não deixem para a última hora!

## Como a feitura será avaliada?

- A pontuação do case vai até 10 pontos, avaliando a lógica por trás do código, a sintaxe do mesmo e o resultado esperado que ele atinja.
- É importante lembrar que o case não é o único elemento avaliado no Processo Seletivo da for\_code. Por esse motivo, sua participação nas outras atividades é de extrema importância.

## ? Mentores e Orientação

- Todo grupo contará com o apoio de um mentor para orientar os trabalhos do time durante a resolução do desafio.
- Utilize o mentor para tirar dúvidas dos exercícios.