<CodeCamp/>
<@FEUP/>

14 fevereiro 2026

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento

acm
Student Chapter

ni

<CodeCamp/>
<@FEUP/>

14
fevereiro
2026

Sessão 3: A etapa final: CEO!

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

Apoio

acm
Student Chapter

ni

<Exercícios/>

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

acm
Student Chapter

ni

A empresa quer **organizar** os seus funcionários de acordo com a **estação do ano** em que nasceram. Para tal, fornecendo-te a **lista de funcionários** e as suas **datas de nascimento**, **retorna** um dicionário com os funcionários **agrupados** por **estação**.

[("Maria Albertina", datetime.datetime(1994, 8, 20)), ("Carlos Mendes", datetime.datetime(2001, 4, 30))]

deverá retornar

{'Inverno': [], 'Primavera': ['Carlos Mendes'], 'Verão': ['Maria Albertina'], 'Outono': []}

Dando seguimento ao exercício anterior, cria uma **função** que receba como **input** a **estação** e retorne a **lista** de funcionários que **fazem anos** nessa **estação** bem como a **o número total de funcionários retornados**.

💡 Dica: O resultado pode ser do tipo tuple(list, int)

Agora, em vez de organizar por estações, agrupa as pessoas que nasceram em anos **bissextos**.

Um ano é **bissexto** se:

- for divisível por 4? **Sim!**
- for divisível por 100 e 4 ao mesmo tempo? **Não!**
- for divisível por 400? **Sim, é uma exceção!**

<Desafio Final/>

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

FEUP
ESCOLAS
Programa de captação
e desenvolvimento de talento.

acm
Student Chapter

ni

<CodeCamp/>
<@FEUP/>

14
fevereiro
2026

Parabéns! Após o teu excelente desempenho nas sessões anteriores, foste finalmente promovido a CEO da empresa. Como CEO, passas a ter acesso a i**nformações confidenciais** que precisam de ser protegidas. Para isso, vais utilizar um dos métodos de **encriptação** mais antigos do mundo - a **Cifra de César**!

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

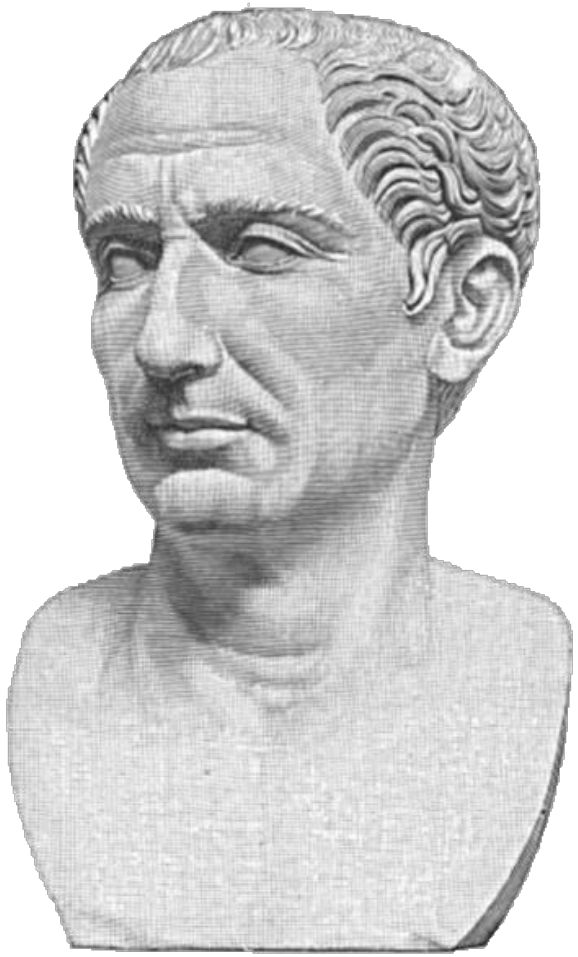
FEUP ESCOLAS
Programa de captação e desenvolvimento de talento.

Apoio

acm Student Chapter

ni

A **Cifra de César** é um dos métodos de encriptação mais antigos do mundo, tendo sido usado pelo próprio **Júlio César**, imperador Romano, para proteger mensagens militares. Esta técnica desloca cada letra do alfabeto um **número fixo** de posições.



Por exemplo, assumindo que o nosso número fixo, a **chave**, é 3:

A -> D

B -> E

C -> F

…

X -> A (**volta ao início!**)

Y -> B

Z -> C

Imaginemos, então, que a mensagem a encriptar é **LUCRO**:

L -> O

U -> X

C -> F

R -> U

O -> R

A nossa mensagem encriptada pela Cifra de César, com chave **3**, passa a ser:

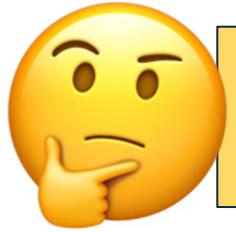
OXFUR

Pedaços de código

```
# Função módulo, que retorna o resto da divisão
print(27 % 26) # resultado = 1
print(30 % 26) # resultado = 4
```

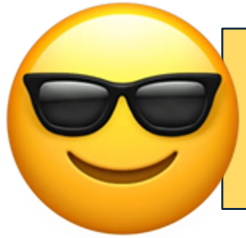
26 letras no alfabeto

```
# ord() converte uma letra para um número, o seu código ASCII
print(ord('A')) # resultado = 65
print(ord('Z')) # resultado = 90
```



Existe um código ASCII diferente para letras minúsculas e maiúsculas

```
# chr() faz o inverso, converte de volta para letra
print(chr(65)) # resultado: 'A'
print(chr(90)) # resultado: 'Z'
```



É mais fácil aplicar a lógica de deslocamento sobre números!

Pedaços de código

```
# Métodos úteis de strings
texto = “Mensagem secreta”
print(texto.upper()) # “MENSAGEM SECRETA”
print(“MENSAGEM”.isAlpha()) # True (verifica se a string apenas contém letras)
print(texto.isAlpha()) # False (neste caso tem um espaço)
```

A tua tarefa é agora, então, fazer uma função de encriptação que aplique estes conceitos. Cria uma função **encriptar(mensagem, chave)** que:

- Recebe uma mensagem e uma chave (o número de deslocamento)
- Encripta apenas as letras (mantendo espaços e pontuação)
- Funciona com maiúsculas e minúsculas, retomando depois a mensagem encriptada

mensagem = encriptar(“Reuniao as 15h”, 5)

print(mensagem) *# "Wjzsnfj fx 15m"*

Dicas:

- Usa **upper()** para maiúsculas
- Verifica se é letra com **isalpha()**
- nova_posicao = (posicao_atual + chave) % 26

Agora, chamando exatamente a mesma função que criaste, como é que podes **desencriptar** uma mensagem?