

Aluno(a): \_\_\_\_\_

## Trabalho 01

- 1) Faça uma função chamada `eh_reveillon` que receba como entrada uma data no formato “aaaa/mm/dd” e verifique se a data é um dia de réveillon, isto é, o último dia do ano. Confira no modo interativo se a função funciona corretamente para os seguintes exemplos.

```
> eh_reveillon("2024/12/31")
True
> eh_reveillon("2000/01/31")
False
> eh_reveillon("2001/12/30")
False
```

- 2) Use o método de substituição e mostre (escreva como comentário) o processo de avaliação para a chamada `eh_reveillon("2000/12/30")`.
- 3) Em um determinado programa é preciso exibir textos em um espaço com uma quantidade fixa de caracteres (assuma que a quantidade de caracteres é pelo menos 5). Se o texto não cabe no espaço, apenas a parte inicial do texto que cabe no espaço seguido de três pontos deve ser exibida. Se falta caracteres, o texto deve ser completado com espaços em branco. Projete uma função que transforme um texto para que ele fique com uma quantidade específica de caracteres conforme o que foi descrito.
- 4) (Desafio) Projete uma função que determine se todos os caracteres de uma string são iguais.

## Referência

```
/// Devolve uma substring de *s* começando em *inicio* e pegando os próximos
/// *tam* caracteres ou até o fim de *s*, o que vier primeiro.
/// Se *inicio* é negativo, começa a partir do fim de *s*.
/// Se *tam* é negativo, devolve "".
```

```
fn string.slice(s: String, inicio: Int, tam: Int) -> Int
```

```
fn string_slice_examples() {
  check.eq(string.slice("funcional", 2, 3), "nci")
  check.eq(string.slice("funcional", 2, -1), "")
  check.eq(string.slice("funcional", 3, 50), "cional")
  check.eq(string.slice("funcional", -3, 2), "na")
}
```

```
/// Cria uma string repetindo *s* o número de *vezes* dado.
/// Se *vezes* é menor ou igual a 0, devolve "".
```

```
fn string.repeat(s: String, vezes: Int) -> String
```

```
fn string_repeat_examples() {
  check.eq(string.repeat("abc", 3), "abcabcabc")
  check.eq(string.repeat("abc", -1), "")
}
```