

Fundamentos de Algoritmos

Conceitos básicos

Marco A L Barbosa

malbarbo.pro.br

Começando

- 1) Indique o valor de cada resultado oculto nessa seção de interações. Use a janela de interações após calcular as respostas manualmente para conferir se elas estão corretas.

```
>>> 2 * 19 // 3
?
>>> 64 ** 1 / 4
?
>>> 15 % 6 * 8 / 4
?
>>> a: int = len('palavra') + 11
>>> a // 2 * 2
?
>>> b: int = a + a // 3
>>> a = 12
>>> b
?
>>> b = 2 * b
>>> b
?
>>> nome: str = 'José da Silva'
>>> nome[:4].upper()
?
>>> nome[5:].lower()
?
>>> nome[3:7]
?
>>> nome[3:7][1]
?
>>> str(len(nome))
?
>>> int('1' * len(nome)) % 100
?
>>> nome = nome + ' e Almeida'
>>> nome
?
```

Praticando

- 2) Faça uma função chamada `area_retangulo` que recebe dois argumentos, a `largura` e a `altura` de um retângulo, e calcula a sua área. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> area_retangulo(3.0, 5.0)
15.0
>>> area_retangulo(2.0, 2.5)
5.0
```

- 3) Faça uma função chamada `produto_anterior_posterior` que recebe um número inteiro `n` e calcula o produto de `n`, `n + 1` e `n - 1`. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> produto_anterior_posterior(3)
24
>>> produto_anterior_posterior(1)
0
>>> produto_anterior_posterior(-2)
-6
```

- 4) Faça uma função chamada `aumenta` que recebe dois número positivos, um `valor` e uma `porcentagem`, e calcula o resultado de aumentar a `porcetagem` ao `valor`. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> aumenta(100.0, 3.0)
103.0
>>> aumenta(20.0, 50.0)
30.0
>>> aumenta(10.0, 80.0)
18.0
```

- 5) Faça uma função chamada `zera_dezena_e_unidade` que recebe um número natural `n` e devolve um novo número que é como `n` mas tem o valor da dezena e unidade zero. Dica: use piso da divisão. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> zera_dezena_e_unidade(19)
0
>>> zera_dezena_e_unidade(341)
300
>>> zera_dezena_e_unidade(5251)
5200
```

- 6) Faça uma função chamada `exclamacao` que recebe dois argumento, uma string `frase` e um número natural `n`, e produz a mesma frase adicionando `n` pontos de exclamação no final da frase. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> exclamacao('Nossa', 3)
'Nossa!!!'
>>> exclamacao('Que legal', 1)
'Que legal!'
>>> exclamacao('Nada', 0)
'Nada'
```

- 7) Faça uma função chamada `primeira_maiuscula` que recebe uma string `frase` e produz a mesma frase mas com a primeira letra em maiúscula. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> primeira_maiuscula('joao venceu.')
'Joao venceu.'
>>> primeira_maiuscula('A Paula é um sucesso.')
'A paula é um sucesso.'
```

- 8) Faça uma função chamada `censura` que recebe dois argumento, uma string `frase` e um número natural `n`, e produz uma nova frase trocando as as primeiras `n` letras da frase de entrada por `n` 'x'. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> censura('droga de lanche!', 5)
'xxxxx de lanche!'
>>> censura('ferrou geral!', 6)
'xxxxxx geral!'
```

- 9) Faça uma função chamada `par` que recebe um número natural `n` e indica se `n` é par. Um número é par se o resto da divisão dele por 2 é igual a zero. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> par(3)
False
>>> par(6)
True
```

- 10) Faça uma função chamada `tres_digitos` que recebe um número natural `n` e verifica se `n` tem exatamente 3 dígitos. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> tres_digitos(99)
False
>>> tres_digitos(100)
True
>>> tres_digitos(999)
True
>>> tres_digitos(1000)
False
```

- 11) Faça uma função chamada `termina_z` que recebe uma string `s` e indica se `s` termina com a letra 'z'. Confira na janela de interações se a função funciona de acordo com os exemplos a seguir

```
>>> termina_z('arroz')
True
>>> termina_z('')
False
>>> termina_z('Z')
True
>>> termina_z('casa')
False
```

Avançando

Para os próximos exercícios, escreva primeiro os exemplos (e deixe como comentários) e depois faça a função.

- 12) Projete uma função que verifique se o caractere do meio de uma string é '-'.
13) Projete uma função que calcule o valor da hipotenusa a partir dos valores dos catetos.
14) Quando escrevemos uma data no Brasil em geral usamos a ordem dia/mês/ano. Já em outros países a ordem usada é mês/dia/ano. Uma outra forma mais recente de escrever a data é ano/mês/dia. Projete uma função que receba como entrada o dia o mês e o ano e gere uma string representando a data na forma ano/mês/dia.
15) Dizemos que o nome de uma pessoa é curto se tem no máximo três letras e longo se tem mais que 8 letras. Um nome que não é nem curto e nem longo é mediano. Projete uma função que verifique se um dado nome é mediano.
16) A operação de resto da divisão é bastante comum na computação, mas muitos ainda não estão acostumados com essa operação, por isso é importante fazermos alguns exemplos para nos familiarizarmos com ela.

- a. Projete uma função que calcule a unidade de um número inteiro positivo, por exemplo, para o número 152, a unidade é 2.
 - b. Projete uma função que calcule a dezena de um número inteiro positivo, por exemplo, para o número 152, a dezena é 5.
 - c. Projete uma função que calcule a centena de um número inteiro positivo, por exemplo, para o número 152, a centena é 1.
 - d. Projete uma função que verifique se os dois últimos dígitos de um número são 00.
- 17) Projete uma função que verifique se o primeiro nome de uma pessoa é “Paula”. Você pode assumir que a string de entrada não tem espaços no início e no final e que contém pelo menos um espaço em branco.
- 18) Projete uma função que determine se três medidas podem formar um triângulo.
- 19) Projete uma função que verifique se o último nome (sobrenome) de uma pessoa é “Silva”. Você pode assumir que a string de entrada não tem espaços no início e no final e que contém pelo menos um espaço em branco.
- 20) (Desafio) Projete uma função que encontre o maior valor entre dois números dados.