

Computação 1, 2021.2

Lista 4

Data limite para entrega: 7/1 às 18:00

Submeta suas soluções colocando os arquivos correspondentes na sua pasta do Google Drive*

Lembre-se de escolher bons nomes para suas funções e variáveis, e de documentar seu código com *docstrings* (documentação de função) e comentários onde for apropriado.

Parte 1 — Obrigatória

Questão 1 (Collatz). A *conjectura de Collatz* afirma que, começando em qualquer natural $n > 0$ e repetindo o processo de trocar n por $3n + 1$ se n for ímpar, ou por $n/2$ se n for par, em algum momento chega-se ao número 1. Por exemplo, começando em $n = 10$, temos a seguinte sequência:

$$10 \mapsto 5 \mapsto 16 \mapsto 8 \mapsto 4 \mapsto 2 \mapsto 1.$$

Não sabemos se a conjectura de Collatz é verdadeira, mas ela já foi verificada para todo $n \leq 2^{68}$.

a. Faça uma função em Python que implemente um “passo” de Collatz, i.e., uma função que receba $n > 0$ inteiro como entrada e retorne o inteiro $3n + 1$ ou o inteiro $n/2$, conforme a condição descrita acima.

b. Faça uma função em Python que “imprima a sequência de Collatz”, no seguinte sentido: a função deve receber $n > 0$ inteiro como entrada, e imprimir na tela cada um dos valores obtidos na sequência de Collatz começando em n , até chegar em 1 (após imprimir 1, sua função deve parar).

Exemplo:

```
>>> nome_da_sua_função(10)
10
5
16
8
4
2
1
```

*Link recebido por email em 24/11/2021 — o nome é parecido com <seu nome> - Comp 1 2021.2 - Submissões e Feedback.

c. Faça uma função em Python que, ao receber $n > 0$ como entrada, retorna o comprimento da sequência de Collatz que começa em n .

Exemplo:

```
>>> nome_da_sua_função(10)
7
```

d. Faça uma função em Python que, ao receber $n > 0$ como entrada, retorna o maior número que aparece na sequência de Collatz que começa em n .

Exemplo:

```
>>> nome_da_sua_função(10)
16
```

e. Faça uma função em Python que, ao receber $n > 0$ como entrada, retorna o maior comprimento dentre as sequências de Collatz que começam em números entre 1 e n (inclusive).

Exemplo:

```
>>> nome_da_sua_função(10)
20
```

Questão 2 (Natação). Nesta questão vamos simular uma prova de natação em piscina olímpica. A piscina tem 8 raias e comprimento de 50 metros. Para fins de simulação, vamos assumir que cada nadador leva um tempo para completar cada *piscina* da prova de acordo com a seguinte regra:

- um *tempo base*, igual a 21 segundos;
- um *tempo de conservação*, igual a (1.5 mais a metade da quantidade total de piscinas da prova) segundos;
- um *tempo aleatório*, entre -1 segundo e +1 segundo.

A sua função deve receber como entrada um inteiro múltiplo de 50, denotando o comprimento total da prova em metros (por exemplo, 800), e deve imprimir na tela a situação da prova a cada *piscina* completa (você deve decidir como essa informação deve ser apresentada na tela). Os números devem ser arredondados a 2 casas decimais (por exemplo, usando a função `round` do módulo `math`), mas não precisa se preocupar em imprimir quem foi o vencedor da prova, etc.

A sua função deve rejeitar entradas que não sejam inteiros positivos múltiplos de 50, apresentando na tela alguma mensagem informativa.

Exemplos:

```
>>> nome_da_sua_função(100):
Após 50m:
Raia 1: 23.15
Raia 2: 23.91
Raia 3: 24.11
Raia 4: 22.59
Raia 5: 23.47
Raia 6: 22.86
Raia 7: 22.53
```

Raia 8: 22.91

Após 100m:

Raia 1: 47.06

Raia 2: 48.05

Raia 3: 46.94

Raia 4: 46.85

Raia 5: 46.18

Raia 6: 47.31

Raia 7: 45.51

Raia 8: 45.82

```
>>> nome_da_sua_função(37):
```

Não é possível fazer uma prova de natação com esse comprimento!

Questão 3 (Euclides). O mdc de números naturais m, n é 0, se $m = n = 0$, ou é o maior número natural que divide *ambos* m e n , caso contrário.

a. Faça uma função que calcule o valor de $\text{mdc}(m, n)$ de forma “ingênua”, diretamente pela definição acima.

b. O valor de $\text{mdc}(m, n)$ pode ser calculado de forma eficiente usando o *Algoritmo de Euclides*: se n for 0, retorne m ; senão, comece dividindo m por n , depois divida n pelo resto da primeira divisão, depois divida o resto da primeira divisão pelo resto da segunda divisão, depois divida o resto da segunda divisão pelo resto da terceira divisão, etc., até que alguma divisão seja exata (i.e., com resto 0). O divisor usado nesta última divisão é o mdc de m e n .

Implemente o algoritmo de Euclides.

Parte 2 — Desafio opcional

Sem desafios essa semana :(