Computação 1, 2021.1
Lista 4

Data limite para entrega: 17/8 às 23:59

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

Lista atualizada em 11/08, adicionando a Questão 3, e em 12/08, corrigindo o exemplo da Questão 3.

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). Use `while` em sua solução.

Exemplo:

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

*Link recebido por email em 19/7/2021 — o nome é parecido com `<seu nome> - Computação 1 2021.1 - 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
```

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). Você não precisa se preocupar em arrendondar números, nem 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: 23.15194467480366
Raia 2: 23.91644496031312
Raia 3: 24.114146999567108
Raia 4: 22.594535501990205
Raia 5: 23.475172139920076
Raia 6: 22.863361590436405
Raia 7: 22.534048086498217
Raia 8: 22.91773455421449

Após 100m:
Raia 1: 47.064983159862855
Raia 2: 48.05360629142507
Raia 3: 46.94412824263277
Raia 4: 46.853112195383346
Raia 5: 46.187737986985255
```

```
Raia 6: 47.318183170202374
Raia 7: 45.51095207678888
Raia 8: 45.82960466674448

>>> 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 $\mathrm{mdc}(m, n)$ de forma "ingênua", diretamente pela definição acima.

b. O valor de $\mathrm{mdc}(m, n)$ 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 :(