Objetivos

- Trabalhar com o funções e parâmetros[1]
- Desenvolver bom estilo de programação

Ficheiros necessários

- LAB5.zip

Exercícios

Verificando Intervalos

Implemente a seguinte função que recebe 3 inteiros como parâmetros:

```
def in_range(n, low, high)
  """
  Retorna True se n está entre low e high, inclusive.
  high é garantidamente maior que low.
  """
```

De seguida, complete o programa escrevendo uma função **`main`** que lê três inteiros do utilizador e chama **`in_range`** para determinar se o segundo número está ou não entre o primeiro e o terceiro. Aqui ficam algumas execuções exemplo do programa (input do utilizador em *itálico*):

Introduza o primeiro número: *42*

Introduza o segundo número: *8*

Introduza o terceiro número: *50*

8 não está entre 42 e 50

Introduza o primeiro número: *8*

Introduza o segundo número: *42*

Introduza o terceiro número: *50*

42 está entre 8 e 50

Introduza o primeiro número: *50*

Introduza o segundo número: *42*

Introduza o terceiro número: *8*

42 está entre 50 e 8

[1] exercícios adaptados de: Mehran Sahami, Chris Piech, Brahm Capoor, Juliette Woodrow, Parth Sarin, Kara Eng, Tori Qiu, Peter Maldonado (2020), Programming Methodologies, Stanford University.

FizzBuzz

No jogo Fizz Buzz, os jogadores contam à vez a partir do número 1. Se a vez do jogador calhar num número divisível por 3, o jogador deve dizer "fizz" em vez do número, e se calhar num número divisível por 5, o jogador deve dizer "buzz" em vez do número. Se o número for um múltiplo de 3 e de 5, o jogador deve dizer "fizzbuzz" em vez do número. Um espetáculo desportivo, não é.

O que é, no entanto, é um problema interessante de controlo de fluxo e utilização de parâmetros. Escreva uma função chamada **`fizzbuzz`** que aceita como parâmetro um inteiro chamado **`n`**. A função deve contar até **`n`** (inclusive), fazendo pelo caminho os fizz e os buzz nos números certos. No final, a função deve retornar quantos números foram "fizzados" ou "buzzados" pelo caminho.

De seguida, complete o seu programa escrevendo uma função **`main`** que lê um inteiro do utilizador e joga fizzbuzz até contar até ao número. Aqui fica um exemplo de uma execução do programa (input do utilizador em *itálico*):

```
Contar até ao número: 17
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
Fizzbuzz
16
17
```

Simulador de Testes Médicos

Uma equipa de médicos e cientistas desenvolveu um novo teste médico para uma doença rara que dizem ter uma precisão de 99%. Contudo, alertam, esse número é enganador. O seu trabalho é escrever um programa que o confirma.

Para sermos exatos, um teste ter uma precisão de 99% significa que para cada 100 pessoas que testa, 99 são corretamente diagnosticadas como saudáveis ou doentes.

De forma a explorar as implicações deste facto, suponha que temos uma população de 10.000 pessoas, das quais 1%, ou 100 pessoas, têm a doença. O seu trabalho é escrever uma função chamada **`simulate_tests`**, que prevê cada uma das seguintes 4 quantidades:

- **O número de verdadeiros positivos**: o número de pessoas doentes que são corretamente diagnosticadas como doentes.
- **O número de falsos positivos:** o número de pessoas saudáveis que são incorretamente diagnosticadas como doentes.
- **O número de falsos negativos:** o número de pessoas doentes que são incorretamente diagnosticadas como saudáveis.
- **O número de verdadeiros negativos:** o número de pessoas saudáveis que são corretamente diagnosticadas como saudáveis.

De forma a podermos generalizar esta previsão a outras doenças e contextos, a sua função deve aceitar os seguintes parâmetros:

```
def simulate_tests(num_people, test_accuracy, infection_rate)
  """
  num_people: o número de pessoas que recebem o teste
  test_accuracy: quão preciso é o teste
  infection_rate: a proporção da população com a doença
  """
```

No exemplo específico discutido acima, a função seria chamada com estes parâmetros:

```
simulate_tests(10000, 0.99, 0.01)
```

A sua função deverá exibir as 4 quantidades mencionadas acima e retornar o rácio de resultados positivos incorretos: ou seja, o número de resultados positivos incorretos dividido pelo número total de resultados positivos.

Este processo pode parecer complexo, por isso vamos dividi-lo em passos. Para simular o teste de uma pessoa, siga os seguintes passos:

1. Escolha aleatoriamente se a pessoa está ou não doente, com uma probabilidade de 0,01.
2. Escolha aleatoriamente se o teste é correto para a pessoa com uma probabilidade de 0,99.
3. Atualize as suas contagens em conformidade.

Poderá querer usar o seguinte excerto de código, que avalia para **`True`** com a probabilidade **`prob`**:

```
is_true = random.random() < prob
```

A função **`random.random()`** retorna um número aleatório entre 0 e 1, cuja probabilidade de ser inferior a **`prob`** é **`prob`**, desde que **`prob`** esteja entre 0 e 1. Por exemplo, um número aleatório entre 0 e 1 tem 50% de hipóteses de ser inferior a 0.5, e assim para obter uma variável que é **`True`** com 50% de probabilidade, usaríamos a seguinte linha:

```
is_true = random.random() < 0.5
```

Depois de escrever esta função, complete o programa escrevendo uma função **`main`** que lê do utilizador um número de pessoas, uma taxa de precisão do teste, e uma taxa de infeção, chama **`simulate_test`**, e de seguida exibe a proporção de resultados positivos incorretos (que o **`simulate_test`** deverá retornar). Pode assumir que o utilizador introduz entradas válidas para cada um destes valores, mas deve decidir de que tipo deverão ser.

Aqui fica um resultado exemplo do programa (input do utilizador em *itálico*):

Número de pessoas: *10000*

Precisão do teste: *0.99*

Taxa de infeção: *0.01*

Verdadeiros positivos: 93

Falsos positivos: 113

Falsos negativos: 1

Verdadeiros negativos: 9793

54.85436893203883% de testes positivos foram incorretos

Como pode constatar, os resultados são bastante surpreendentes: perto de 55% das pessoas que foram informadas que tinham a doença na verdade não a tinham. Intuitivamente, porque é que isto será assim? Será que o nosso resultado é apenas anómalo?