

# Teste FPLN

## 8 Questions

1. Considere a seguinte função **fun1**, e diga se a seguinte afirmação é verdadeira ou falsa:

A função **fun1** recebe uma lista, e devolve uma nova lista com os mesmos elementos da lista recebida, mas sem repetições.

```
def fun1(l: list) -> list:
    res = []
    for x in l:
        if x not in res:
            res.append(x)
    return res
```

13/16 ☒ True

3/16 ☐ False

2. Considere a seguinte função **mdc**.  
Escolha as afirmações verdadeiras:

```
def mdc(x, y):
    if y == 0:
        return x
    else:
        return mdc(y, x % y)
```

- 1/16 ☐ A O resultado de **mdc(14, 24)** é 8. 4/16 ☐ B Nesta função, a ordem dos argumentos influencia o seu resultado. Isto é, para quaisquer **a** e **b** inteiros: **mdc(a,b) != mdc(b,a)**

14/16 ☒ C A função **mdc** é uma função recursiva.

4/16 ☐ D A seguinte imagem mostra uma função equivalente à função **mdc** original.

```
def mdc(x, y):
    while y > 0:
        r = x % y
        x = y
        y = r
    return x
```

3. Considere a seguinte variável:

**A** = [[1,2,3],  
[4,5,6],  
[7,8,9]]

Qual das seguintes instruções resultará no output: **6**?

0/16 ☐ A **A[2][1]**

4/16 ☐ B **A[2][3]**

9/16 ☒ C **A[1][2]**

3/16 ☐ D **A[3][2]**

4. Imagine que foi fazer compras a uma loja que tem a seguinte política de venda:

Numa compra acima de 100€, poderá utilizar um cupão que oferece 10% de desconto (se o tiver!).

Escolha quais das seguintes opções poderão ser uma implementação do cálculo do preço total de uma compra de uma pessoa nesta loja:

14/15 **A**

```
def preco_total(preco_compra, tem_cupao):
    if preco_compra > 100:
        if tem_cupao:
            return preco_compra * 0.9
        else:
            return preco_compra
    else:
        return preco_compra
```

11/15 **B**

```
def preco_total(preco_compra, tem_cupao):
    if preco_compra > 100 and tem_cupao:
        return preco_compra * 0.9
    else:
        return preco_compra
```

1/15 **C**

```
def preco_total(preco_compra, tem_cupao):
    if preco_compra > 100 or tem_cupao:
        return preco_compra * 0.9
    return preco_compra
```

0/15 **D**

```
def preco_total(preco_compra, tem_cupao):
    resultado = preco_compra
    if preco_compra >= 100:
        resultado = preco_compra
    if tem_cupao:
        resultado = preco_compra * 0.9
    return resultado
```

5. Qual o resultado da seguinte expressão:  
`len([1, 2, "", True, ['a', 'b','c']])`

0/16 **A** 4

0/16 **B** 2

4/16 **C** Erro

12/16 **D** 5

6. Considere o seguinte dicionário **jogo**.  
Assinale as afirmações verdadeiras:

5/15 **A** A seguinte função devolve uma lista com os minutos de todos os acontecimentos que ocorreram no jogo.

```
def minutos_acontecimentos(jogo):
    r = []
    for a in jogo['acontecimentos']:
        if a['tipo'] == 'golo':
            r.append(a['minuto'])
    return r
```

11/15 **B** A expressão `jogo['acontecimentos'][1]['interveniante']['equipa']` devolve o valor "FC Porto"

```
jogo = {
    "competição": "Campeonato",
    "equipa-casa": "FC Porto",
    "equipa-fova": "SL Benfica",
    "resultado": "3-1",
    "acontecimentos": [
        {
            "tipo": "golo",
            "minuto": 34,
            "interveniante": {
                "nome": "Fábio Vieira",
                "equipa": "FC Porto"
            }
        },
        {
            "tipo": "golo",
            "minuto": 37,
            "interveniante": {
                "nome": "Papé",
                "equipa": "FC Porto"
            }
        },
        {
            "tipo": "golo",
            "minuto": 46,
            "interveniante": {
                "nome": "Yaremchuk",
                "equipa": "SL Benfica"
            }
        },
        {
            "tipo": "expulsão",
            "minuto": 49,
            "interveniante": {
                "nome": "André Almeida",
                "equipa": "SL Benfica"
            }
        },
        {
            "tipo": "golo",
            "minuto": 89,
            "interveniante": {
                "nome": "Ivan",
                "equipa": "FC Porto"
            }
        }
    ]
}
```

13/15 **C** Uma possível forma de descobrir o nome da equipa vencedora, seria percorrer os acontecimentos do jogo, e ir contando os acontecimentos do tipo "golo" de cada equipa. No fim, caso houvesse desigualdade nos contadores, devolveria o nome da equipa com mais golos. Caso fossem iguais, poderia devolver "Empate".

1/15 **D** O número de acontecimentos do jogo pode ser obtido através da seguinte expressão:  
`len(jogo['acontecimentos'][0])`

7. Diga se a seguinte afirmação é verdadeira ou falsa:

A expressão `[0, 'a', 1, 'b', 2, 'c', 3][::2]`  
retorna `['a', 'b', 'c']`

5/16 ☐ True

11/16 ☒ False

8. Considere a seguinte expressão **e**:

**e = x + y**

Selecione as afirmações verdadeiras:

0/16 ☐ A Se **x = "1"** e **y = "20"** o valor de **e** é **"21"**.

10/16 ☐ B Se **x = "1"** e **y = "20"** o valor de **e** é **"120"**.

16/16 ☒ C Se **x = 1** e **y = 20** o valor de **e** é **21**.

1/16 ☐ D Se **x = int("1")** e **y = 20** o programa dará erro.