

**Gabriel Fernandes Bispo - 823143267**

### **Atividade 1 – Função maior(a, b)**

Objetivo: Criar uma função que retorne o maior entre dois números.

#### **Implementação:**

```
def maior(a, b):  
    return a if a > b else b
```

#### **Testes:**

```
def test_maior_errado():  
    assert maior(5, 3) == 3 # Falha proposital
```

```
def test_maior_forcado():  
    assert maior(5, 5) == 5 # Forçado
```

```
def test_maior_correto():  
    assert maior(10, 7) == 10 # Correto
```

### **Atividade 2 – Função inverter\_string(texto)**

Objetivo: Criar uma função que devolve a string invertida.

#### **Implementação:**

```
def inverter_string(texto: str) -> str:  
    return texto[::-1]
```

#### **Testes:**

```
def test_inverter_errado():  
    assert inverter_string("abc") == "abc" # Falha proposital
```

```
def test_inverter_forcado():  
    assert inverter_string("x") == "x" # Forçado
```

```
def test_inverter_correto():  
    assert inverter_string("python") == "nohtyp" # Correto
```

### Atividade 3 – Função valida\_cpf(cpf)

Objetivo: Criar uma função que verifique se a string possui 11 dígitos numéricos.

#### Implementação:

```
def valida_cpf(cpf: str) -> bool:  
    return cpf.isdigit() and len(cpf) == 11
```

#### Testes:

```
def test_valida_cpf_errado():  
    assert valida_cpf("12345678901") is False # Falha proposital
```

```
def test_valida_cpf_forçado():  
    assert valida_cpf("00000000000") is True # Forçado
```

```
def test_valida_cpf_correto():  
    assert valida_cpf("12345") is False # Correto
```