

Laço fechado - Correção

NUMWORKS

1 Exercício

Escreva a função `soma(n)` que tome um número inteiro como argumento e devolva a soma dos n primeiros inteiros ($1 + 2 + 3 + \dots + n$) e que contenha um laço `for`.

1.1 Análise do enunciado

Pretendemos utilizar um laço `for` na função `soma(n)`. Portanto, antes de mais, é necessário identificar a operação repetitiva e determinar quantas vezes é repetida.

À mão, podemos começar por testar os casos $n=1$, $n=2$, $n=3$, ...

- $n = 1$:

$$S(1) = 1$$

- $n = 2$:

$$S(2) = 1 + 2 = 3$$

- $n = 3$:

$$S(3) = 1 + 2 + 3 = 6$$

Continuando com $n=4$, $n=5$, ... vamos facilmente perceber que é melhor adicionar 4, 5, ... ao resultado anterior em vez de recalculá-lo tudo: preferimos assim fazer $S(4) = S(3) + 4 = 6 + 4 = 10$ do que $S(4) = 1 + 2 + 3 + 4$.

Acabámos de distinguir então a operação que se repete: " $S(i) = S(i-1) + i$ ".

Para calcular a soma dos números inteiros de 1 a n , é necessário fazer variar i de 1 a n . Utilizaremos então `range(1, n+1)`.

1.2 Resolução

Tomamos S como a soma procurada. No início do laço, a variável S será 0: $S = 0$. No final do laço, queremos $S = 1 + 2 + \dots + n$.

```
gra PYTHON
1 def soma(n):
2     S=0
3     for i in range(1,n+1):
4         S=S+i
5     return S
6
7
8
9
10
11
12
```

Antes do laço `for`, a variável `S` é criada, começando por 0. A cada iteração do laço `for`, é executada a seguinte operação: “armazena `S+i` em `S`”.

1.3 Uma nota para terminar

Existe uma fórmula explícita em função de `n` para calcular a soma dos primeiros `n` números inteiros:

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

2 Outro exercício

Escreva uma função `potencia(x,n)` que tome um valor real `x` e um inteiro natural `n` como argumento e devolva a `x` elevado `n`-ésima potência, x^n .

2.1 Correção

2.2 Algumas notas sobre a correção

Como repetimos `n` vezes $x \times x \times \dots \times x$, utilizamos o laço `for` bem como a soma dos `n` primeiros números inteiros. Aqui multiplicamos o resultado por `x` a cada iteração.

```
gra PYTHON
>>> 2**3
8
>>> |
```

```
gra PYTHON
1 def potencia(x,n):
2     resultado=1
3     for i in range(n):
4         resultado=resultado*x
5     return resultado
6
7
8
9
10
11
12
```

Figura 1: *

image 2

Em Python, há a possibilidade de utilizar o duplo asterisco para representar a potência.

```
gra PYTHON
>>> from math import *
>>> pow(2,3)
8.0
>>> |
```

Além disso, o módulo `math` de Python dispõe da função `pow(x,n)` para calcular potências.