

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Computational Thinking

PROF. EDUARDO GONDO

Funções - Exemplos

Escreva um algoritmo que recebe um inteiro positivo n e calcula $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n$. Por exemplo, se $n = 6$, então $6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$.

```
1  n = int(input("Informe n: "))
2  prod = 1
3  while n >= 1:
4      prod = prod * n
5      n = n - 1
6
7  print(prod)
```

```
1  def fatorial(n):
2      prod = 1
3      while n >= 1:
4          prod = prod * n
5          n = n - 1
6
7      return prod
8
9  fat = fatorial(6)
10 print(fat)
```

Funções - Exemplos

Dizemos que um inteiro positivo n é perfeito se for igual a soma de seus divisores positivos diferentes de n . Escreva um algoritmo que dado n inteiro positivo, verifica se ele é perfeito.

```
1  n = int(input("Informe n: "))
2  div = 1
3  soma = 0
4
5  while div < n:
6      if n % div == 0:
7          soma = soma + div
8          div = div + 1
9
10 if n == soma:
11     print("Eh perfeito")
12 else:
13     print("Nao eh perfeito")
14
15 def perfeito(n):
16     div = 1
17     soma = 0
18
19     while div < n:
20         if n % div == 0:
21             soma = soma + div
22             div = div + 1
23
24     if n == soma:
25         return True
26     else:
27         return False
28
29 resp = perfeito(6)
30 print(resp)
```

Funções - Exemplos

Dado um número inteiro na base decimal, converta para sua representação binária.

```
1  n = int(input("Informe n: "))
2  pot = 1
3  soma = 0
4
5  while n != 0:
6      resto = n % 2
7      soma = soma +
          resto * pot
8      pot = pot * 10
9      n = n // 2
10
11 print(soma)
```

```
1  def decToBin(n):
2      pot = 1
3      soma = 0
4
5      while n != 0:
6          resto = n % 2
7          soma = soma +
              resto * pot
8          pot = pot * 10
9          n = n // 2
10
11     return soma
12
13
14 resp = decToBin(6)
15 print(resp)
```

Funções - Exemplos

- ▶ Dados dois números inteiro positivos a e b , escreva um algoritmo e uma função que encontra o menor número inteiro que é múltiplo do número a e do número b .
- ▶ Dados dois números inteiro positivos a e b , escreva um algoritmo e uma função que encontra o maior número inteiro que divide o número a e o número b .

Funções - Exercícios

- ▶ Escreva uma função que dados dois números reais, retorna a média geométrica.
- ▶ Escreva uma função que dado um inteiro positivo n , verifica se n é primo ou não.
- ▶ Um número a é dito permutação de um número b se os dígitos de a formam uma permutação dos dígitos de b .
Exemplo: 5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455. Obs.: Considere que o dígito 0 (zero) não aparece nos números.
 - (a) Faça uma função `contadigitos` que dados um inteiro n e um inteiro d , $0 \leq d \leq 9$, devolve quantas vezes o dígito d aparece em n .
 - (b) Usando a função do item anterior, faça um programa que lê dois inteiros positivos a e b e responda se a é permutação de b .

Funções - Exercícios

- ▶ Construa uma função encaixa que dados dois inteiros positivos a e b verifica se b corresponde aos últimos dígitos de a. veja

<i>a</i>	<i>b</i>	
567890	890	=> encaixa
1243	1243	=> encaixa
2457	245	=> não encaixa
457	2457	=> não encaixa

um exemplo:

- ▶ Usando a função do item anterior, faça um programa que lê dois inteiros positivos a e b e verifica se o menor deles é segmento do outro.

<i>a</i>	<i>b</i>	
567890	678	=> b é segmento de a
1243	2212435	=> a é segmento de b
235	236	=> um não é segmento do outro

Referência Bibliográfica

- ▶ Puga e Rissetti - Lógica de Programação e Estrutura de Dados
- ▶ Ascêncio e Campos - Fundamentos da Programação de Computadores
- ▶ Forbelone e Eberspacher - Lógica de programação: a construção de algoritmos e estruturas de dados
- ▶ Documentação do Python - <https://docs.python.org/3.8/>
- ▶ Python Programming For Beginners: Learn The Basics Of Python Programming (Python Crash Course, Programming for Dummies) (English Edition). Kindle
- ▶ Python: 3 Manuscripts in 1 book: - Python Programming For Beginners - Python Programming For Intermediates - Python Programming for Advanced (English Edition). Kindle

| Copyleft

Copyleft © 2025 Prof. Eduardo Gondo Todos direitos liberados.
Reprodução ou divulgação total ou parcial deste documento é liberada.