

Departamento de Engenharia de Sistemas e Computação

Exercícios e Revisão

Revisão geral dos capítulos 1 a 4 (preparação para a Prova 1)

João Araujo

Algoritmos Computacionais I

Aula 14

Mapa da matéria da Prova 1

Capítulo	Tópicos principais	Aulas
1. Computadores	bases 2/8/10/16, complemento a 2, IEEE 754, ASCII, von Neumann, compilado *vs.* interpretado	2, 3
2. Bases	algoritmo, fluxograma, pseudocódigo, identificadores, tipos, constantes	4, 5, 6
3. Estruturada	`if/elif/else`, `while`, `for`, laços encaixados	7, 8, 9
4. Subalgoritmos	funções, parâmetros, escopo, módulos, `try/except`	10 a 13

Cap. 1: o que você precisa saber fazer

- Converter entre **decimal, binário, octal e hexadecimal** (à mão)
- Representar inteiros com sinal em **complemento a 2** e somar em binário
- Explicar por que $0{,}1 + 0{,}2 \neq 0{,}3$ (IEEE 754)
- Diferença entre **kilo** e **kibi**; calcular tempos de download e capacidades
- Desenhar o modelo de **von Neumann** e explicar CPU, memória, E/S
- Diferenciar linguagens **compiladas**, **interpretadas** e **semi-interpretadas**

Exercícios do site (cap. 1)

1.3 a 1.9 (conversões), 1.10, 1.11 (unidades), 1.12 a 1.15 (complemento a 2), 1.16 a 1.18 (reais)

Mini-teste: Capítulo 1

1. Converta 237_{10} para binário e hexadecimal.
2. Qual a faixa de inteiros representáveis com 12 bits em complemento a 2?
3. Escreva -18 em complemento a 2 com 8 bits.
4. Quantos MiB cabem em um arquivo de 5 000 000 bytes? (1 MiB $= 2^{20}$ bytes)
5. Por que `0.1 + 0.2 == 0.3` dá `False`?

Respostas: (1) $11101101_2 = \text{ED}_{16}$; (2) -2048 a 2047; (3) `1110 1110`; (4) $\approx 4{,}77$ MiB; (5) 0,1 e 0,2 não têm representação binária exata.

Cap. 2: o que você precisa saber fazer

- Escrever um **algoritmo** em pseudocódigo e desenhar o **fluxograma**
- Decidir se um nome é um **identificador** válido
- Reconhecer tipos: `int`, `float`, `str`, `bool`, `complex`
- Usar `input` com conversão de tipo e `print` formatado
- Calcular expressões respeitando a **precedência**: `**`, `*`, `/`, `//`, `%`

Mini-teste: o que cada linha imprime?

```
print(7 // 2, 7 % 2, 7 / 2, 2 ** 3 ** 2, "5"+ "3", int("5") + 3)
```

Resposta: `3 1 3.5 512 53 8`

Cap. 3: leia o código e diga a saída

```
soma = 0
for i in range(1, 6):
    if i % 2 == 0:
        continue
    soma += i
print(soma)

n = 20
while n > 0:
    n -= 7
print(n)
```

- Primeiro laço: soma dos ímpares de 1 a 5 $\rightarrow$ **9**
- Segundo: $20 \rightarrow 13 \rightarrow 6 \rightarrow -1 \rightarrow$ **-1**
- Faça o "teste de mesa": uma tabela com os valores das variáveis a cada volta

Cap. 3: escreva o código

1. Leia um inteiro $n > 0$ e mostre a soma dos **múltiplos de 3** entre 1 e n.
2. Leia notas até o usuário digitar -1; mostre a média e quantas notas foram ≥ 7.
3. Escreva a condição "o ano a é bissexto".
4. Imprima o padrão (laços encaixados):

```
1
1 2
1 2 3
1 2 3 4
```

Soluções comentadas de problemas semelhantes: exercícios 3.7 a 3.15 do site.

Cap. 4: leia o código (escopo e parâmetros)

```
x = 10

def f(a, b=2):
    x = a * b
    return x

def g():
    print(x)

print(f(3), f(3, 4), f(b=5, a=1))
g()
print(x)
```

- `f(3)` $\rightarrow$ 6; `f(3, 4)` $\rightarrow$ 12; `f(b=5, a=1)` $\rightarrow$ 5
- `g()` imprime **10**: o `x` de `f` é local
- Último `print`: **10**

Cap. 4: escreva o código

1. Função `eh_primo(n)` que devolve `True` ou `False`.
2. Função `estatisticas(a, b, c)` que devolve o **menor**, o **maior** e a **média** dos três números.
3. Função `le_inteiro_positivo(msg)` que só aceita inteiros > 0 (use `try/except`).

```
def eh_primo(n):
    if n < 2:
        return False
    for d in range(2, int(n ** 0.5) + 1):
        if n % d == 0:
            return False
    return True
```

Teste: `eh_primo(1)`, `eh_primo(2)`, `eh_primo(97)`, `eh_primo(91)`.

Erros que mais custam nota

- Esquecer de converter o resultado de `input`
- = no lugar de ==
- `range(n)` versus `range(1, n+1)`
- Acumulador que não é reiniciado
- Dividir sem proteger o zero
- `print` no lugar de `return`
- Usar variável local fora da função
- Recuo (indentação) inconsistente
- Comparar `float` com ==
- Não testar os limites das faixas

Orientações para a Prova 1 (Aula 15)

- **Conteúdo:** capítulos 1 a 4 do livro
- Parte teórica à mão (conversões, complemento a 2, teste de mesa)
- Parte de programação: escrever funções e programas completos
- Estude refazendo os exercícios do site *sem* olhar a solução
- Use os exercícios adicionais para treinar mais:
  `https://www.araujo.eng.uerj.br/livro/extras/`

Prova 1

Aula 15: sem consulta. Chegue com tempo e releia o enunciado duas vezes.

Lista de exercícios para revisão (site do livro)

Capítulo 1	1.3 a 1.9, 1.10, 1.11, 1.12 a 1.18
Capítulo 2	2.1 a 2.8, 2.9 a 2.18
Capítulo 3	3.1 a 3.15
Capítulo 4	4.1 a 4.17 (pelo menos 4.7, 4.8, 4.11, 4.12, 4.13, 4.16)

`https://www.araujo.eng.uerj.br/livro/exercicios/`