

6 Reunião 06: 17/SET/2020

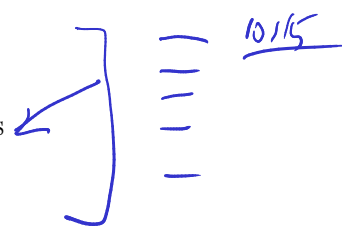
6.1 Aulas passadas

- padrão para iterar um bloco n vezes $i = 0$, while $i < n$: ... $i=i+1$: *fatorial*
- contadores: *fatorial* com contador de iterações e *conta pares e ímpares*
- divisão inteira $//$: *conta pares e ímpares*
- resto de divisão $%$: *conta pares e ímpares*
- execução condicional if: *conta pares e ímpares*
- execução alternativa if-else: *conta pares e ímpares*

Python

6.2 Avisos

- provinha 03
- exercícios na seção Exercícios
- "Test drives" dos EPs
- EP03



EPs
ideias

Fazer exercícios, não precisa daqueles que envolvem os operadores and, or, not para a provinha.

6.3 Hoje

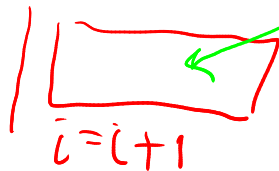
- mais execução condicional if: *frequência dígito*
- mais operadores // e '%': *dígitos*
- execução alternativa if-else: *estatística de notas*
- execução em cadeia if-elif-else: *estatística de notas e ~~multiplos de i e j~~*
- desenvolver e testar programas em partes: *estatística de notas onde calculamos primeiro no número de reprovados, depois o número de recuperação,...*

Nota: o EP03 é um mistura do fatorial e do frequência dígito

Ideias

→ padrão para repetir n vezes

$i = 0$
while $i < n$:



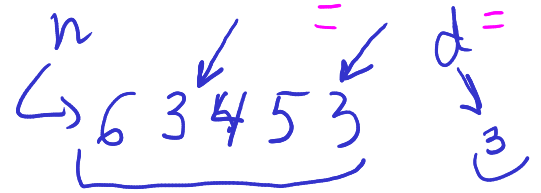
bloco de comandos a serem repetidos

→ fazer os pedaços simples — leitura
— impressão
— inicialização de contadores

→ fazer e testar o programa: passo a passo um pouco por vez

6.4 Exercício: frequência de dígito

Dados um número inteiro n , $n > 0$, e um dígito d ($0 \leq d \leq 9$), determinar quantas vezes d ocorre em n .
 Por exemplo, 3 ocorre 2 vezes em 63453.

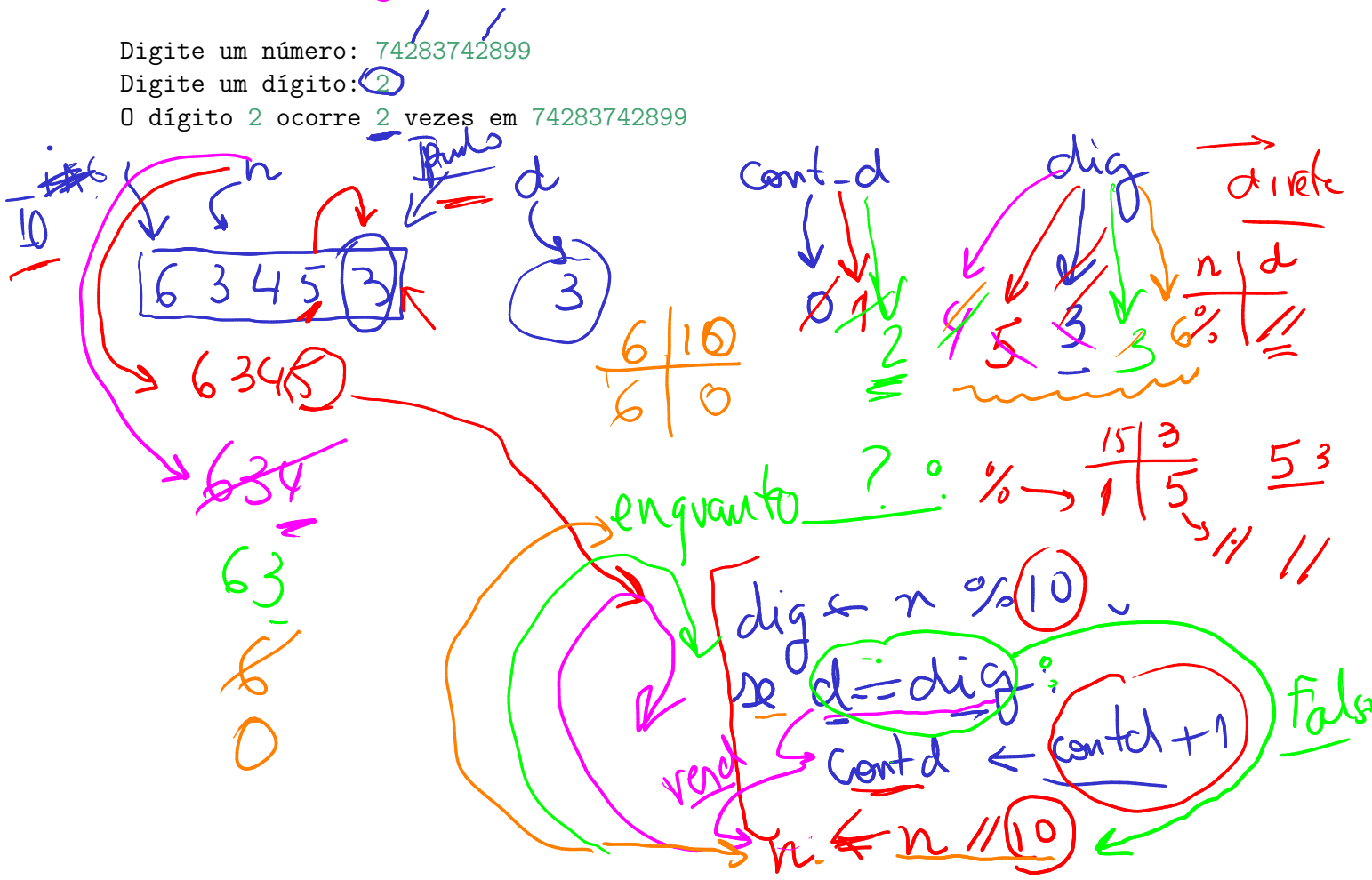


6.4.1 Exemplos

Digite um número: 316253644
 Digite um dígito: 0
 O dígito 0 ocorre 0 vezes em 316253644

Digite um número: 7834724239484823
 Digite um dígito: 3
 O dígito 3 ocorre 3 vezes em 7834724239484823

Digite um número: 74283742899
 Digite um dígito: 2
 O dígito 2 ocorre 2 vezes em 74283742899



6.4.3 Solução

```
n = int(input("Digite um número: "))  
d = int(input("Digite um dígito: "))
```

```
conta_dig = 0
```

```
n_salvo = n
```

```
while n > 0:
```

```
    dig = n % 10
```

```
    n = n // 10
```

```
    if dig == d:
```

```
        conta_dig = conta_dig + 1
```

```
print(f"O dígito {d} ocorre {conta_dig} vezes em {n_salvo}")
```

Tópicos: esse problema serve como revisão para //, %, if e contador e tem umas ideias legais.

1192 // 3

EP03

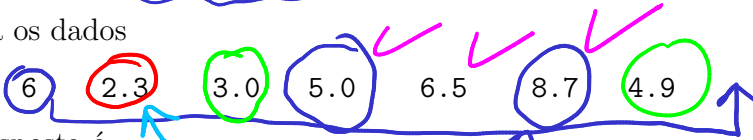
6.5 Exercício: notas finais

Dados um número inteiro n , $n > 0$, e as notas finais de n estudantes, determinar o:

1. número reprovad(a|o)s ($nf < 3$); ✓
2. número de recuperação ($3 \leq nf < 5$); ✓
3. número aprovad(a|o)s ($nf \geq 5$); ✓
4. número bam-bam-bam ($nf \geq 8$). ✓

Python } taideiro
 rep | rec | apr | bca

Para os dados



A resposta é

Total de alunos = 6

Número reprovad(a|o)s = 1

Número de recuperacao = 2

Número aprovad(a|o)s = 3

Número bam-bam-bam = 1

$i = 0$
 while $i < 10$ {
 [scribble]
 }
 $i = i + 1$

6.5.1 Exemplo

Digite o número de estudantes: 6

Digite uma nota final: 2.3

Digite uma nota final: 3

Digite uma nota final: 5

Digite uma nota final: 6.5

Digite uma nota final: 8.7

Digite uma nota final: 4.9

Total estudantes= 6

Número reprovad(a|o)s = 1

Número de recuperacao = 2

Número aprovad(a|o)s = 3

Número bam-bam-bam = 1

6.5.3 Solução parcial 1

```
n = int(input("Digite o número de estudantes: "))
no_repr = 0

i = 0
while i < n:
    nf = float(input("Digite uma nota final: "))
    n = n - 1
    if nf < 3:
        no_repr = no_repr + 1
    i = i + 1

print("-----")
print("Total de estudantes=", n)
print("Número reprovad(a|o)s =", no_repr)
```

6.5.4 Solução parcial 2

```
n = int(input("Digite o número de estudantes: "))
no_repr = 0
no_recu = 0

i = 0
while i < n:
    nf = float(input("Digite uma nota final: "))
    n = n - 1
    if nf < 3:
        no_repr = no_repr + 1
    else if nf < 5:
        no_recu = no_recu + 1
    i = i + 1

print("-----")
print("Total de estudantes=", n)
print("Número reprovad(a|o)s =", no_repr)
print("Número de recuperacao =", no_recu)
```


6.5.5 Solução parcial 3

Quem quiser pode tentar explorar combinações com vários ifs, if-elses, e if-elifs.

```
n = int(input("Digite o número de estudantes: "))
no_repr = 0
no_recu = 0
no_apro = 0

i = 0
while n > 0:
    nf = float(input("Digite uma nota final: "))
    n = n - 1
    if nf < 3:
        no_repr = no_repr + 1
    elif nf < 5:
        no_recu = no_recu + 1
    else:
        no_aprov = no_aprov + 1
    i = i + 1

print("-----")
print("Total de estudantes=", n)
print("Número reprovad(a|o)s =", no_repr)
print("Número de recuperacao =", no_recu)
print("Número aprovad(a|o)s =", no_aprov)
```

6.6 Solução

```
no_aprov = 0
no_recu = 0
no_repr = 0
no_bam = 0

i = 0
while i < n:
    nota_final = float(input("Digite uma nota final: "))
    if nota_final >= 5.0:
        no_aprov += 1
        if nota_final >= 8.0:
            no_bam += 1
    elif nota_final >= 3.0:
        no_recu += 1
    else:
        no_repr += 1
    i = i + 1

print("-----")
print("Total de estudantes=", n)
print("Número reprovad(a|o)s =", no_repr)
print("Número de recuperacao =", no_recu)
print("Número aprovad(a|o)s =", no_aprov)
print("Número bam-bam-bma =", no_bam)
```

6.7 Execução condicional em cadeia

```
if condição_0:
    comandos_0;
elif condição_1:
    comandos1;
elif condição_2:
    comandos_2
...
elif condição_k
    comandos_k
else:
    comandos_k+1
```

6.7.1 Significado

As condições são testadas em ordem, uma após outra. Apenas os comandos correspondente a primeira condição True será executada. Os comandos associados ao else serão executados apenas se todas as condições forem falsas.

```
n = int(input("Digite um número: "))
if n < 0:
    print(f"{n} é negativo")
elif n == 0:
    print(f"{n} é zero")
elif n == 1:
    print(f"{n} é um")
else:
    print(f"{n} é maior que um")
```

Tópicos: função float() e execução condicional if, alternativa if-else e em cadeia if-elif-else. Podem haver sugestões de soluções que usam operadores lógicos and' eor. O operador lógico not' parece ser mais raro.

6.8 Execução condicional em cadeia

```
if condição_0:
    comandos_0;
elif condição_1:
    comandos1;
elif condição_2:
    comandos_2
...
elif condição_k
    comandos_k
else:
    comandos_k+1
```

6.8.1 Significado

As condições são testadas em ordem, uma após outra. Apenas os comandos correspondente a primeira condição True será executada. Os comandos associados ao else serão executados apenas se todas as condições

forem falsas.

```
n = int(input("Digite um número: "))
if n < 0:
    print(n, " é negativo.")
elif n == 0:
    print(n, " é zero.")
elif n == 1:
    print(n, " é um.")
else:
    print(n, "é maior que um.")
```