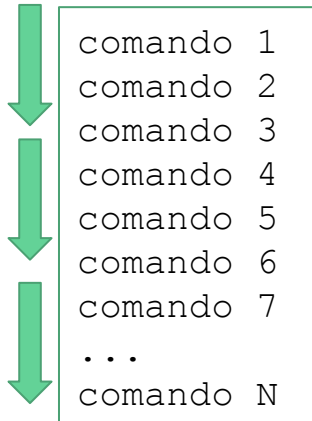


# MC102 - Algoritmos e Programação de Computadores

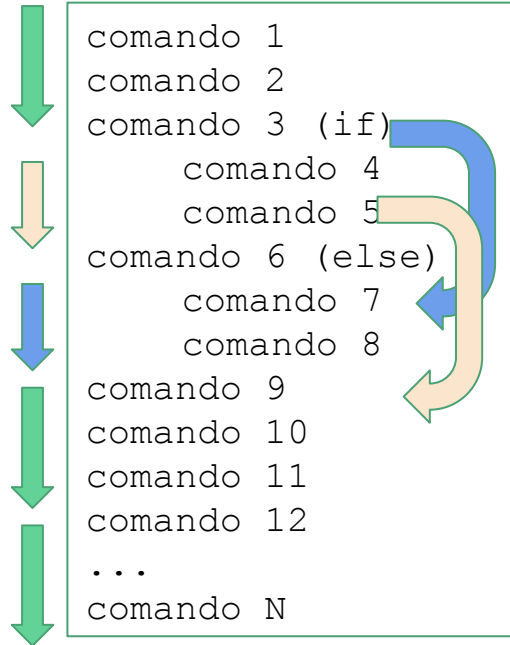
---

Turma Z - Segundo Semestre de 2019

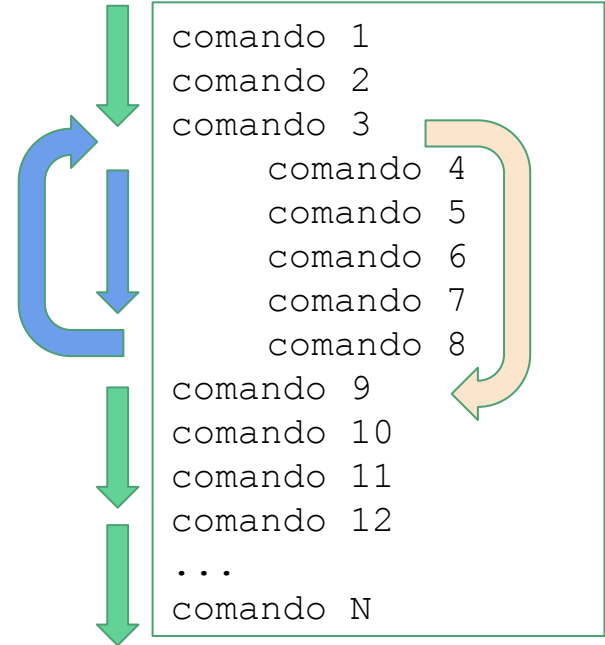
# 1ª semana



# 2ª semana



# Agora



A partir desse slide, os slides com fundo cinza claro são parte do material desenvolvido pela professora Sandra Avila e disponível em <http://www.ic.unicamp.br/~sandra/>

# Resumo

- O uso de variáveis **acumuladora**, **indicadora** e **contadora** são úteis em várias situações.
- Mas não existem fórmulas para a criação de soluções para problemas.
- Em outros problemas, o uso destes padrões pode aparecer em conjunto, ou nem mesmo aparecer como parte da solução.

# Exercício: Corrida de Lesmas (Lab. 2018/1)



<https://www.ic.unicamp.br/~mc102/mc102-1s2018/labs/roteiro-lab05.html>



# Corrida de Lesmas

- A tarefa de capturar lesmas velozes não é uma tarefa muito fácil, pois praticamente todas as lesmas são muito lentas. Cada lesma é classificada em um nível dependendo de sua velocidade:
  - **Nível 1:** Se a velocidade é menor que 10 cm/h.
  - **Nível 2:** Se a velocidade é maior ou igual a 10 cm/h e menor que 20 cm/h.
  - **Nível 3:** Se a velocidade é maior ou igual a 20 cm/h.
- Sua tarefa é identificar qual nível de velocidade da lesma mais veloz de um grupo de lesmas.

# Corrida de Lesmas



- A **entrada** consiste várias linhas:
  - A primeira linha contém um inteiro  $L$  ( $1 \leq L \leq 100$ ) representando o número de lesmas do grupo.
  - As outras  $L$  linhas contêm inteiros  $V_i$  ( $1 \leq V_i \leq 50$ ) representando as velocidades de cada lesma do grupo.

# Corrida de Lesmas

- Para a saída, imprima uma única linha indicando o nível de velocidade da lesma mais veloz do grupo.
- Caso algum dos valores esteja fora dos intervalos estabelecidos, uma mensagem de erro deve ser emitida indicando a linha em que o erro ocorreu:

Valor inválido na linha <l>.







# Corrida de Lesmas

Entrada	Saída
5	2
12	
9	
8	
7	
6	

Entrada	Saída
10	1
1	
5	
2	
9	
5	
5	
8	
4	
4	
3	

Entrada	Saída
5	Valor
10	inválido
90	na linha
8	3.
7	
6	

```
lesmas = int(input("Digite o número de lesmas do grupo: "))
maior = 0
erro = False

if (1 <= lesmas <= 100): # condição do número de lesmas
    for linha in range(lesmas):
        velocidade = int(input())
        if (1 <= velocidade <= 50): # condição da velocidade da lesma
            if (maior < velocidade):
                maior = velocidade
            else:
                print("Valor inválido na linha " + str(linha+2) + ".")
                erro = True

        if (not erro):
            if (maior < 10):
                print("A lesma mais veloz está no nível 1.")
            elif (maior < 20):
                print("A lesma mais veloz está no nível 2.")
            else:
                print("A lesma mais veloz está no nível 3.")
    else:
        print("Valor inválido na linha 1.")
```

# Agenda

---

- Laços encaixados
- Exercício

# Laços Encaixados

- Para resolver alguns problemas, é necessário implementar um laço dentro de outro laço.
- Estes são laços encaixados.

```
for i in range(1,11):  
    for j in range(1,6):  
        print(i, j)
```

- O que será impresso por este programa?

# Laços Encaixados

```
for i in range(1,11):  
    for j in range(1,6):  
        print(i, j)
```

- Fixado um valor para **i** no primeiro laço **for**, começa-se o segundo laço **for**, que varia o valor de **j** entre 1 e 5.

# Laços Encaixados

```
for i in range(1,11):  
    for j in range(1,6):  
        print(i, j)
```

- Fixado um valor para **i** no primeiro laço **for**, começa-se o segundo laço **for**, que varia o valor de **j** entre 1 e 5.
- No final deste segundo laço **for**, voltamos para o primeiro laço onde a variável **i** assumiria seu próximo valor. Fixado este valor de **i** começa-se novamente o segundo laço **for**.

# Laços Encaixados

```
for i in range(1,11):  
    for j in range(1,6):  
        print(i, j)
```

# Laços Encaixados

```
for i in range(1,11):  
    for j in range(1,6):  
        print(i, j)
```

```
1 1  
1 2  
1 3  
1 4  
1 5  
2 1  
...  
10 5
```



# Laços Encaixados

- Um uso comum de laços encaixados ocorre quando para cada um dos valores de uma determinada variável, precisamos gerar/checar algo com valores de outras variáveis.

# Laços Encaixados: Equações Lineares

- Determinar todas as soluções inteiras de um sistema linear como:

$$x_1 + x_2 = C$$

com  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $C \geq 0$  e todos inteiros.

- Uma solução: para cada um dos valores de  $0 \leq x_1 \leq C$ , teste todos os valores de  $x_2$  possíveis e verifique quais deles são soluções.

Para cada  $x_1$  entre 0 e C faça

Para cada  $x_2$  entre 0 e C faça

Se  $x_1 + x_2 = C$  então imprima solução

# Laços Encaixados: Equações Lineares

Para cada  $x_1$  entre 0 e C faça

Para cada  $x_2$  entre 0 e C faça

Se  $x_1 + x_2 = C$  então imprima solução

```
C = int(input("Digite o valor da constante C: "))

for x1 in range(0,C+1):
    for x2 in range(0,C+1):
        if (x1 + x2 == C):
            print(x1, " + ", x2, " = ", C)
```

# Laços Encaixados: Equações Lineares

- **Observação:** Note que fixado  $x_1$ , não precisamos testar todos os valores de  $x_2$ , pois este é determinado como  $x_2 = C - x_1$ .

```
C = int(input("Digite o valor da constante C: "))  
  
for x1 in range(0,C+1):  
    x2 = C - x1  
    print(x1, " + ", x2, " = ", C)
```

- Mas, no caso geral com  $n$  variáveis,  $x_1 + x_2 + x_3 + x_4 + \dots + x_n = C$ , será preciso fixar  $(n - 1)$  variáveis só para determinar o valor de  $x_n$ .

# Laços Encaixados: Equações Lineares

- Determinar todas as soluções inteiras de um sistema linear como:

$$x_1 + x_2 + x_3 = C$$

com  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $x_3 \geq 0$ ,  $C \geq 0$  e todos inteiros.

Para cada  $x_1$  entre 0 e C faça

    Para cada  $x_2$  entre 0 e C faça

        Para cada  $x_3$  entre 0 e C faça

            Se  $x_1 + x_2 + x_3 = C$  então imprima solução

# Laços Encaixados: Equações Lineares

Para cada  $x_1$  entre 0 e C faça

    Para cada  $x_2$  entre 0 e C faça

        Para cada  $x_3$  entre 0 e C faça

            Se  $x_1 + x_2 + x_3 = C$  então imprima solução

```
C = int(input("Digite o valor da constante C: "))

for x1 in range(0,C+1):
    for x2 in range(0,C+1):
        for x3 in range(0,C+1):
            if (x1 + x2 + x3 == C):
                print(x1, " + ", x2, " + ", x3, " = ", C)
```

# Laços Encaixados: Equações Lineares

- Note que fixado  $x_1$ , não precisamos testar todos os valores de  $x_2$ , pois o valor máximo de  $x_2$  é  $C - x_1$ .
- Fixados  $x_1$  e  $x_2$ , o valor de  $x_3$  é determinado como  $C - x_1 - x_2$ .

```
C = int(input("Digite o valor da constante C: "))

for x1 in range(0,C+1):
    for x2 in range(0,C+1):
        for x3 in range(0,C+1):
            x3 = C - x1 - x2
            print(x1, " + ", x2, " + ", x3, " = ", C)
```

# Exemplo

- Faça um programa que leia um número  $n$  e imprima  $n$  linhas na tela com o seguinte formato (exemplo se  $n = 5$ ):

Entrada	Saída
5	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5



# Exemplo

- Faça um programa que leia um número  $n$  e imprima  $n$  linhas na tela com o seguinte formato (exemplo se  $n = 5$ ):

```
n = int(input())

for i in range(1, n+1):
    for j in range(1, i+1):
        print(j, end=" ")
    print()
```

# Exercício

- Faça um programa que leia um número  $n$  e imprima  $n$  linhas na tela com o seguinte formato (exemplo se  $n = 5$ ):

Entrada	Saída
5	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1

```
n = int(input())
```

```
for i in range(1,n+1):  
    for j in range(1,i+1):  
        print(j, end=" ")  
    print()
```

Entrada	Saída
5	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5

Entrada	Saída
5	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1

# Exercício

```
n = int(input())

for i in range(1, n+1):
    for j in range(1, i+1):
        print(j, end=" ")
    print()

for i in range(n-1, 0, -1):
    for j in range(1, i+1):
        print(j, end=" ")
    print()
```

# Exercício

Imprima o primeiro número primo maior que  $Y$  para:

- $Y = 3$
- $Y = 20$
- $Y$  qualquer.

Postulado de Bertrand: Para  $n$  natural maior que 1, existe ao menos um número primo entre  $n$  e  $2n$

# Mais Exercícios

- Faça um programa que peça 10 números inteiros, calcule e mostre a quantidade de números pares e a quantidade de números ímpares.
- Numa eleição existem três candidatos. Faça um programa que peça o número total de eleitores. Peça para cada eleitor votar e ao final mostrar o número de votos de cada candidato.

Em python, o tipo de uma variável é determinado pelo valor que ela recebe no momento de sua atribuição. Até o momento, vimos variáveis dos tipo inteiro ('int'), ponto flutuante ('float'), string ('str') e booleanos ('bool').

Determine o **valor** e o **tipo** de cada variável abaixo:

```
a = 3
b = 2.0
c = 3 / 2
d = 3 // 2
e = a + b
f = a // b
g = '2.00'
h = float(f)
i = int(b)
j = format(b, '.2f')
k = True
l = a == 3
m = a == 3 and b == 2
n = a == 3 and a != 3
o = g == 2.00
```

Obs: a função `type()` permite verificar o tipo de uma variável. Utilize esta função para verificar suas respostas. Ex:

```
>>> a = 3
>>> type(a)
<class 'int'>
```

## Exercício Extra =)

\*Elaborado por Lucas de Magalhães Araújo

# Referências & Exercícios

- Os slides dessa aula foram baseados no material de MC102 do Prof. Eduardo Xavier (IC/Unicamp)
- <https://wiki.python.org.br/EstruturaDeRepeticao>: 51 exercícios \o/
- <https://panda.ime.usp.br/pensepy/static/pensepy/07-Iteracao/maisiteracao.html>