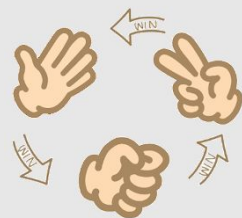
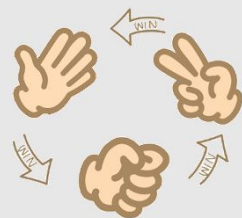


# Exercício: Pedra, Papel e Tesoura



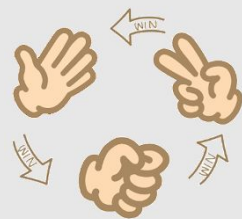
- Vamos continuar o programa “Pedra, Papel e Tesoura”.
  - O jogador só pode digitar 0 (pedra), 1 (papel) ou 2 (tesoura). Imprima a mensagem “Opção inválida” se não for nenhuma dessas opções.
  - Vamos jogar novamente? Se “Sim”, recomece o jogo. Se “Não”, encerre o jogo.

# Exercício: Pedra, Papel e Tesoura



- Vamos continuar o programa “Pedra, Papel e Tesoura”.
  - O jogador só pode digitar 0 (pedra), 1 (papel) ou 2 (tesoura). Imprima a mensagem “Opção inválida” se não for nenhuma dessas opções.
  - Vamos jogar novamente? Se “Sim”, recomece o jogo. Se “Não”, encerre o jogo.

# Exercício: Pedra, Papel e Tesoura

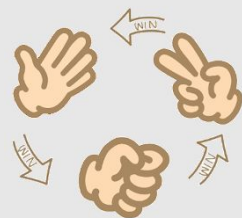


```
jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))
jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))

pedra = 0
papel = 1
tesoura = 2

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.")
elif (jogador1 - jogador2 == -2) or (jogador1 - jogador2 == 1):
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

# Exercício: Pedra, Papel e Tesoura

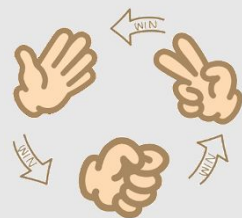


```
jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))
jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))

pedra = 0
papel = 1
tesoura = 2

if (0 <= jogador1 <= 2) and (0 <= jogador2 <= 2):
    if (jogador1 == jogador2):
        print("Empate! Ninguém ganhou.")
    elif (jogador1 - jogador2 == -2) or (jogador1 - jogador2 == 1):
        print("Jogador 1 ganhou.")
    else:
        print("Jogador 2 ganhou.")
else:
    print("Opção inválida.")
```

# Exercício: Pedra, Papel e Tesoura



- Vamos continuar o programa “Pedra, Papel e Tesoura”.
  - O jogador só pode digitar 0 (pedra), 1 (papel) ou 2 (tesoura). Imprima a mensagem “Opção inválida” se não for nenhuma dessas opções.
  - Vamos jogar novamente? Se “Sim”, recomece o jogo. Se “Não”, encerre o jogo.

```
pedra = 0
papel = 1
tesoura = 2

jogar_novamente = "Sim"

while (jogar_novamente == "Sim"):

    jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: " ))
    jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: " ))

    if (0 <= jogador1 <= 2) and (0 <= jogador2 <= 2):
        if (jogador1 == jogador2):
            print("Empate! Ninguém ganhou.")
        elif (jogador1 - jogador2 == -2) or (jogador1 - jogador2 == 1):
            print("Jogador 1 ganhou.")
        else:
            print("Jogador 2 ganhou.")
    else:
        print("Opção inválida.")

    jogar_novamente = input("Você quer tentar novamente? Digite Sim ou Não" )

print("Até a próxima!")
```



# Algoritmos e Programação de Computadores

Comandos Repetitivos: Laços Encaixados

**Profa. Sandra Avila**

Instituto de Computação (IC/Unicamp)

MC102, 28 Março, 2018

# Agenda

---

- Laços encaixados
- Exercício
- Lab01: Minissomadora



# 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.
- 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)
```

```
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-x1):
        x3 = C - x1 - x2
        print(x1, " + ", x2, " + ", x3, " = ", C)
```

# Exercício

# 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

# Exercício

- 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

# 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("")
```

# Lab01: Minissomadora



# Lab01: Minissomadora

- Você deverá implementar um programa em Python que soma dois números. Cada número virá precedido de uma letra **i** ou **f** indicando se este é um `int` ou `float`, como esquematizado a seguir:

```
<tipo_operando_1>  
<valor_operando_1>  
<tipo_operando_2>  
<valor_operando_2>
```

- Conforme o tipos dos operandos, o resultado será um `float` formatado com exatamente duas casas decimais ou um `int`.

# Lab01: Minissomadora

Entrada	Saída
i 5 i 3	8

Entrada	Saída
i 3 f 3.57	6.57

Entrada	Saída
f 5.58 i 4	9.58

Entrada	Saída
f 2.3589 f 6.3234	8.68

# Lab01: Minissomadora

```
tipol = input() # i para inteiro ou f para float
if tipol == "i":
    numero1 = int(input())
else:
    numero1 = float(input())

tipo2 = input() # i para inteiro ou f para float
if tipo2 == "i":
    numero2 = int(input())
else:
    numero2 = float(input())

if (type(numero1) == int and type(numero2) == int):
    print(numero1 + numero2) # número inteiro
else:
    print(format(numero1 + numero2, ".2f")) # número float
```

# Lab01: Minissomadora

```
tipol = input() # i para inteiro ou f para float
if tipol == "i":
    numero1 = int(input())
else:
    numero1 = float(input())

tipo2 = input() # i para inteiro ou f para float
if tipo2 == "i":
    numero2 = int(input())
else:
    numero2 = float(input())

if (type(numero1) == float or type(numero2) == float):
    print(format(numero1 + numero2, ".2f")) # número float
else:
    print(numero1 + numero2) # número inteiro
```

# Lab01: Minissomadora

```
tipol = input() # i para inteiro ou f para float
if tipol == "i":
    numero1 = int(input())
else:
    numero1 = float(input())

tipo2 = input() # i para inteiro ou f para float
if tipo2 == "i":
    numero2 = int(input())
else:
    numero2 = float(input())

if (tipol == "i" and tipo2 == "i"):
    print(numero1 + numero2) # número inteiro
else:
    print(format(numero1 + numero2, ".2f")) # número float
```

# Lab01: Minissomadora

```
tipol = input() # i para inteiro ou f para float
if tipol == "i":
    numero1 = int(input())
else:
    numero1 = float(input())

tipo2 = input() # i para inteiro ou f para float
if tipo2 == "i":
    numero2 = int(input())
else:
    numero2 = float(input())

if (tipol == "f" or tipo2 == "f"):
    print(format(numero1 + numero2, ".2f")) # número float
else:
    print(numero1 + numero2) # número inteiro
```

# Lab01: Minissomadora

```
tipol = input() # i para inteiro ou f para float
if tipol == "i":
    numero1 = int(input())
else:
    numero1 = float(input())

tipo2 = input() # i para inteiro ou f para float
if tipo2 == "i":
    numero2 = int(input())
else:
    numero2 = float(input())

resultado = numero1 + numero2
if (type(resultado) == int):
    print(resultado) # número inteiro
else:
    print(format(resultado, ".2f")) # número float
```

# 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/
- Curso de Python:
  - <https://www.codecademy.com/learn/learn-python>