

O que vimos na aula passada?

Exemplo: Números em Ordem Crescente

- Faça um programa que lê n números inteiros do teclado, e no final informa se os números lidos estão ou não em ordem crescente.

Entrada	Saída
5	Sim
2	
4	
13	
22	
41	

Entrada	Saída
4	Não
6	
3	

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

i = 1 # leu um número
ordenado = True # ordenado é a variável indicadora

while (i < n) and (ordenado):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual < anterior):
        ordenado = False
    anterior = atual

if (ordenado):
    print("Sequência está ordenada.")
else:
    print("Sequência não está ordenada.")
```

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

i = 1 # leu um número
ordenado = True # ordenado é a variável indicadora

while (i < n) and (ordenado):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual < anterior):
        ordenado = False
    anterior = atual

if (ordenado):
    print("Sequência está ordenada.")
else:
    print("Sequência não está ordenada.")
```

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

i = 1 # leu um número
ordenado = 0 # ordenado é a variável contadora

while (i < n) and (ordenado == 0):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual < anterior):
        ordenado = ordenado + 1
        anterior = atual

if (ordenado == 0):
    print("Sequência está ordenada.")
else:
    print("Sequência não está ordenada.")
```

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

i = 1 # leu um número
ordenado = 0 # ordenado é a variável contadora

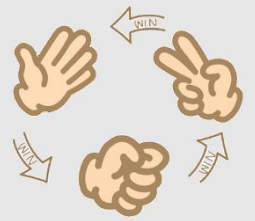
while (i < n) and (ordenado == 0):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual < anterior):
        ordenado = ordenado + 1
    anterior = atual

if (ordenado == 0):
    print("Sequência está ordenada.")
else:
    print("Sequência não está ordenada.")
```

Resumo

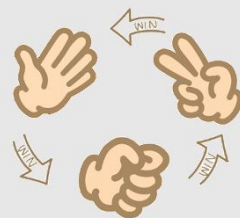
- O uso de variáveis **acumuladora**, **indicadora** e **contadora** são úteis em várias situações.
- Mas não existem fórmulas para a criação de soluções para problemas.
- Em outros problemas, o uso destes padrões pode aparecer em conjunto, ou nem mesmo aparecer como parte da solução.

Exercício: 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.

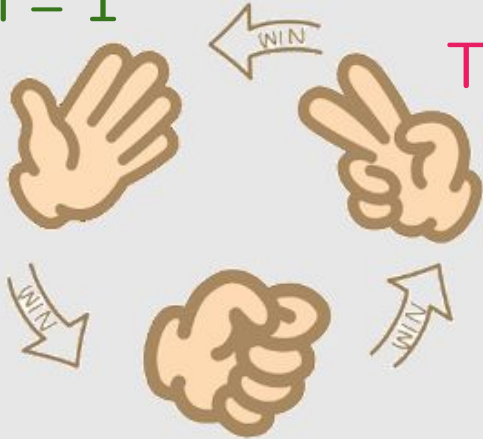
Exercício: Pedra, Papel e Tesoura



- 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

Papel = 1



Tesoura = 2

Pedra = 0

$$(0 - 2) \% 3 = 1 \text{ (Ganhou)}$$

$$(2 - 1) \% 3 = 1 \text{ (Ganhou)}$$

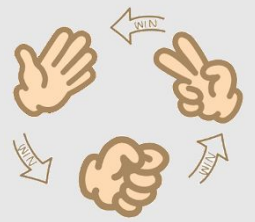
$$(1 - 0) \% 3 = 1 \text{ (Ganhou)}$$

$$(0 - 1) \% 3 = 2$$

$$(1 - 2) \% 3 = 2$$

$$(2 - 0) \% 3 = 2$$

Exercício: Pedra, Papel e Tesoura

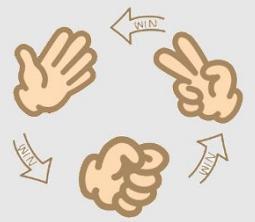


```
jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))
jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))

pedra = 0
papel = 1
tesoura = 2

if (jogador1 == jogador2):
    print("Empate! Ninguém ganhou.") # empate
elif (jogador1 - jogador2) % 3 == 1:
    print("Jogador 1 ganhou.")
else:
    print("Jogador 2 ganhou.")
```

Exercício: Pedra, Papel e Tesoura

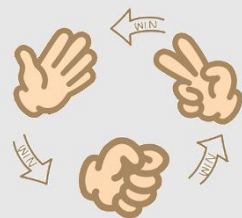


```
jogador1 = int(input("Jogador1, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))
jogador2 = int(input("Jogador2, digite 0 p/pedra, 1 p/papel ou 2/tesoura: "))

pedra = 0
papel = 1
tesoura = 2

if (0 <= jogador1 <= 2) and (0 <= jogador2 <= 2):
    if (jogador1 == jogador2):
        print("Empate! Ninguém ganhou.") # empate
    elif (jogador1 - jogador2) % 3 == 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.") # empate
        elif (jogador1 - jogador2) % 3 == 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!")
```

Exercício: Número Adjacente

- Faça um programa que lê n números inteiros do teclado, e no final informa se os números lidos tem dois dígitos adjacentes iguais.

Exemplos:

Para $n = 5$ números inteiros e 21212, a resposta é não.

Para $n = 5$ números inteiros e 21221, a resposta é sim.

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

i = 1 # leu um número
nao_adjacente = True # nao_adjacente é a variável indicadora

while (i < n) and (nao_adjacente):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual == anterior):
        nao_adjacente = False
    anterior = atual

if (nao_adjacente):
    print("Não, não é adjacente.")
else:
    print("Sim, é adjacente.")
```



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

i = 1 # leu um número
nao_adjacente = True # nao_adjacente é a variável indicadora

while (i < n) and (nao_adjacente):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual == anterior):
        nao_adjacente = False ← continue
    anterior = atual

if (nao_adjacente):
    print("Não, não é adjacente.")
else:
    print("Sim, é adjacente.")
```

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

i = 1 # leu um número
nao_adjacente = True # nao_adjacente é a variável indicadora

while (i < n):
    atual = int(input())
    i = i + 1 # leu mais um número
    if (atual == anterior):
        nao_adjacente = False
        continue
    anterior = atual

if (nao_adjacente):
    print("Não, não é adjacente.")
else:
    print("Sim, é adjacente.")
```

Exercício: IMC normal?

- Dado como entrada a **altura** e o **peso** de uma pessoa, construa um programa que calcula o IMC (Índice de Massa Corporal).
- Por exemplo, uma pessoa com 1,70 m e 70 kg fará o seguinte cálculo:
 - IMC é igual ao peso dividido pelo quadrado da altura.
 - $IMC = 70 / (1,70 \times 1,70) = IMC = 24,22$

IMC	Resultado
Menor que 18,6	Abaixo do peso
Entre 18,6 e 24,9	Peso normal
Entre 25 e 29,9	Sobrepeso
Maior que 29,9	Obesidade

```
altura = float(input("Informe a sua altura (em metros): "))
peso = float(input("Informe o seu peso (em kg): "))

imc = peso / (altura*altura)
if (imc < 18.6):
    print("Você está abaixo do peso ideal.")
elif (18.6 <= imc <= 24.9):
    print("Você está no seu peso ideal.")
elif (25 <= imc <= 29.9):
    print("Você está com sobrepeso.")
else:
    print("Você está obeso.")
```

```
altura = float(input("Informe a sua altura (em metros): "))
peso = float(input("Informe o seu peso (em kg): "))

imc = peso / (altura*altura)
if (imc < 18.6):
    print("Você está abaixo do peso ideal. IMC =", format(imc, '.2f'))
elif (18.6 <= imc <= 24.9):
    print("Você está no seu peso ideal. IMC =", format(imc, '.2f'))
elif (25 <= imc <= 29.9):
    print("Você está com sobrepeso. IMC =", format(imc, '.2f'))
else:
    print("Você está obeso. IMC =", format(imc, '.2f'))
```

```
altura = float(input("Informe a sua altura (em metros): "))
peso = float(input("Informe o seu peso (em kg): "))

imc = peso / (altura*altura)
if (imc < 18.6):
    print("Você está abaixo do peso ideal.", end=" ")
elif (18.6 <= imc <= 24.9):
    print("Você está no seu peso ideal.", end=" ")
elif (25 <= imc <= 29.9):
    print("Você está com sobrepeso.", end=" ")
else:
    print("Você está obeso.", end=" ")

print("IMC =", format(imc, '.2f'))
```

```
altura = 0
peso = 0

while (peso <= 0 or altura <= 0):
    altura = float(input("Informe a sua altura (em metros): "))
    peso = float(input("Informe o seu peso (em kg): "))

imc = peso / (altura*altura)
if (imc < 18.6):
    print("Você está abaixo do peso ideal.", end=" ")
elif (18.6 <= imc <= 24.9):
    print("Você está no seu peso ideal.", end=" ")
elif (25 <= imc <= 29.9):
    print("Você está com sobrepeso.", end=" ")
else:
    print("Você está obeso.", end=" ")

print("IMC =", format(imc, '.2f'))
```



Algoritmos e Programação de Computadores

Comandos Repetitivos: Laços Encaixados

Profa. Sandra Avila

Instituto de Computação (IC/Unicamp)

MC102, 27 Março, 2019

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
```

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



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

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>