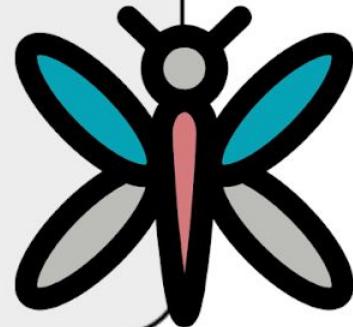




# **Projeto "Libélulas": Treinamento para OBI Modalidade Programação**



Núcleo de Estudos e Pesquisa em Tecnologia da Informação



```
010011101101100111000010010000001001110010001010101  
00000110010101010001001001001000010010000100100001
```



**INSTITUTO FEDERAL**  
Goiano

Campus  
Ceres



Vai começar...

*Boas-vindas!*





OBI

*Programação esportiva e divertida!*



# Ranking atualizado!



**Façam as listas e  
participem do fórum  
para ganhar pontos!**



 [bit.ly/ranking\\_libelulas](https://bit.ly/ranking_libelulas)



# Estrutura Condicional

## Parte 01

*Aula 03*





# Executar ou não executar?!

- **Nem sempre** todas as linhas dos programas serão executadas. Em algumas **situações**, será necessário **decidir** que partes **vão ser executadas**, com base em uma **condição**;
  - Para auxiliar nesse processo, fazemos uso também de **operações lógicas e relacionais**.



0101010101010000101001010101010101000010101110111010001110001001010101010101010100  
0101010101010100001010010101010101010100001010111011101000111000100101010101010100



# A estrutura condicional

- As **condições** servem para **selecionar** quando uma **parte do programa** deve ser **ativada** e quando deve ser simplesmente ignorada;
- A **estrutura condicional** é representado pelo “**IF**”, também conhecido como “**SE**”.

IF...  
...

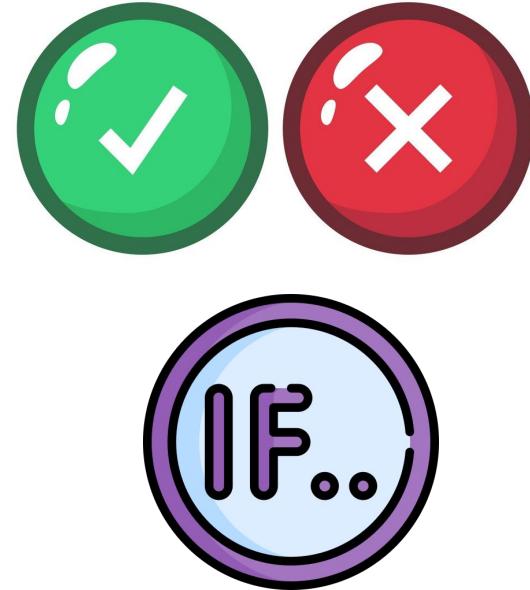


0101010101010000101001010101010100001010110111010001110001001010101010101010100  
010101010101000010100101010101010000101011101110100011100010010101010101010100



# A estrutura condicional

- Em **Python**, a estrutura de decisão é o **IF**.  
Seu formato é representado a seguir:
    - **if <condição>:**  
**bloco verdadeiro**
  - Podemos então, **entendê-lo em português** da seguinte forma: se a **condição for verdadeira, faça algo.**



0101010101010000101001010101010101000010101110111010001110001001010101010101010100  
0101010101010100001010010101010101010100001010111011101000111000100101010101010100



# O chapéu seletor em Python!

- Em **condicionais**, devemos relembrar dos **operadores relacionais**, como no exemplo a seguir:

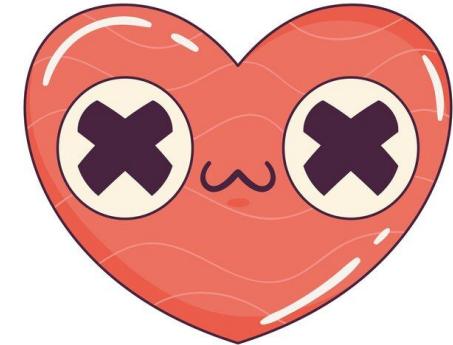
```
chapeu = "Grifinória";
if chapeu == "Grifinória":
    print("Você é da Grifinória")
if chapeu == "Sonserina":
    print("Você é da Sonserina")
```





## A cláusula ELSE

- Para alguns **contextos**, pode ser necessário **executar** trechos de código em que **situação** é a **inversa da primeira**;
  - Essa forma é por meio do comando **“ELSE”**, para especificar o que fazer caso o resultado da **avaliação da condição** caso seja **falso**.



```
01010101010101000010100101010101010100001010111011101000111000100101010101010101010100  
010101010101010000101001010101010101000010101110111010001110001001010101010101010100
```



# Como usar o ELSE?!

- Para usar o “ELSE”, siga essa estrutura:

```
idade = int(input("Digite a idade: "))

if idade >=18:
    print("Você é maior de idade!")

else:
    print("Você não é maior de idade!")
```





# A cláusula ELIF

- A cláusula “**ELIF**” permite que outras condições sejam testadas, sobretudo além da condição inicial “**IF**”;
  - O “**ELIF**” é usado para criar **condições alternativas** em uma **estrutura condicional if-else**, permitindo que o código siga **caminhos diferentes**.



```
01010101010100001010010101010100001010111010001110001001010101010101010100  
01010101010101000010100101010101010000101011101000111000100101010101010100
```



# Usando a cláusula ELIF!

```
numero = int(input("Digite um número: "))
```

**if** numero > 0:

```
print("O número é positivo.")
```

**elif** numero < 0:

```
print("O número é negativo.")
```

**else:**

```
print("O número é zero.")
```

**Assim, todas as condições serão testadas!**



0101010101010000101001010101010101000010101110111010001110001001010101010101010100  
0101010101010100001010010101010101010100001010111011101000111000100101010101010100



# Vamos para prática juntos!

beecrowd | 1050

## DDD

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Leia um número inteiro que representa um código de DDD para discagem interurbana. Em seguida, informe à qual cidade o DDD pertence, considerando a tabela abaixo:

DDD	Destination
61	Brasilia
71	Salvador
11	Sao Paulo
21	Rio de Janeiro
32	Juiz de Fora
19	Campinas
27	Vitoria
31	Belo Horizonte

Se a entrada for qualquer outro DDD que não esteja presente na tabela acima, o programa deverá informar:  
DDD nao cadastrado

**Exercício da  
Lista 03 já  
disponível!**

010101010101000010100101010101010000101011011010001110001001010101010101010100  
0101010101010000101001010101010100001010110110100011100010010101010101010100



# Vamos para prática juntos!

```
ddd = int(input())
if ddd == 61:
    print("Brasilia")
elif ddd == 71:
    print("Salvador")
elif ddd == 11:
    print("Sao Paulo")
elif ddd == 21:
    print("Rio de Janeiro")
elif ddd == 32:
    print("Juiz de Fora")
elif ddd == 19:
    print("Campinas")
elif ddd == 27:
    print("Vitoria")
elif ddd == 31:
    print("Belo Horizonte")
else:
    print("DDD nao cadastrado")
```



0101010101010000101001010101010101000010101110111010001110001001010101010101010100  
010101010101010000101001010101010101010000101011101110100011100010010101010101010100



# Agora é hora da OBI!

*Aula 03*





# Desafio - Agora é hora da OBI!

## Ogro

*Nome do arquivo: ogro.c, ogro.cpp, ogro.java, ogro.js ou ogro.py*

Ogro e Bicho-Papão têm fama de malvados, mas na verdade são amáveis, honestos e trabalhadores, além de vizinhos e amigos. O Bicho-Papão tem dificuldades em aprender aritmética e por isso o Ogro inventou uma brincadeira simples para auxiliar seu amigo: o Ogro inicia mostrando um certo número de dedos na sua mão esquerda (vamos chamar esse valor de  $E$ ) e um número de dedos diferente na mão direita (vamos chamar esse valor de  $D$ ). Então, Bicho-Papão deve falar o *resultado* da brincadeira, definido assim:

- se o número de dedos na mão esquerda é maior do que o número de dedos na mão direita (ou seja  $E > D$ ) então o resultado é a soma dos dois números (ou seja  $E + D$ );
- caso contrário, o resultado é o dobro da diferença entre o número de dedos na mão direita e o número de dedos na mão esquerda (ou seja,  $2 \times (D - E)$ ).

O problema é que o Ogro também não é lá muito bom em aritmética, e pediu sua ajuda para conferir se o Bicho-Papão falou a resposta correta.

Dados o número de dedos mostrados na mão esquerda ( $E$ ) e o número de dedos mostrados na mão direita ( $D$ ), escreva um programa para determinar a resposta da brincadeira.



01010101010100001010010101010101000010101101100011000100101010101010101010100  
01010101010101000010100101010101010000101011011000110001001010101010101010100



# Questão - O Ogro - OBI 2024

## Entrada

A entrada é composta por duas linhas. A primeira linha contém um inteiro  $E$ , o número de dedos mostrados na mão esquerda. A segunda linha contém um inteiro  $D$ , o número de dedos mostrados na mão direita.

## Saída

Seu programa deve produzir uma única linha na saída, contendo um único número inteiro, o resultado da brincadeira.



## Exemplos

Exemplo de entrada 1	Exemplo de saída 1
1 0	1

01010101010100001010010101010101010000101011011010001110001001010101010101010100  
010101010101000010100101010101010100001010110110100011100010010101010101010100



# Resolução - O Ogro - OBI 2024

```
e = int(input())
d = int(input())

if e > d:
    res = e + d
else:
    res = 2 * (d - e)

print(res)
```



**OBI 2024**

010101010101000010100101010101010100001010110111010001110001001010101010101010100  
01010101010100001010010101010101010000101011101110100011100010010101010101010100



## Dúvidas e perguntas?!

-  @nepeti.ce no Instagram  
[obi.ce@ifgoiano.edu.br](mailto:obi.ce@ifgoiano.edu.br)
-  [informatica.ifgoiano.edu.br](http://informatica.ifgoiano.edu.br)
-  Fórum de dúvidas (Moodle)



TREINANDO EM CASA  
**Listas 01, 02 e 03**  
no Beecrowd :)

Dúvidas no fórum  
do moodle!





# Material Licenciado



Este documento está licenciado sob uma licença Creative Commons CC BY-NC-SA 4.0.

Texto da licença: <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.pt-br>

*Esta licença permite que outros remixem, adaptem e criem a partir do seu trabalho para fins não comerciais, desde que atribuam a você o devido crédito e que licenciem as novas criações sob termos idênticos.*



**Atribuição** — Você deve dar o crédito apropriado, prover um link para a licença e indicar se mudanças foram feitas.



**NãoComercial** — Você não pode usar o material para fins comerciais.



**Compartilhagual** — Se você remixar, transformar, ou criar a partir do material, tem de distribuir as suas contribuições sob a mesma licença que o original.



01010100  
01001001

NEPeTI