Problema: verificar se um número introduzido pelo utilizador é par

Nota: usar condição if (não usar if else)

Inicio

n

N%2=0

“Par”

**Em Python ciclo If**

true

Fim

n= input('Digite um numero: ')

if int(n)%2==0:

print('Par')

* Em Python para o ciclo If não existe o ‘then’ e não existe o ‘end’ para terminar o ciclo, o compilador sabe o que está dentro e fora do ciclo pela maneira como o código está identado. É sempre necessário o uso dos ‘:’ antes do bloco de comandos

**if** <condição>**:**

<instruções>

**Em Lua ciclo If**

io.write'Digite um número:'

n = io.read()

if n%2==0 then

print 'Par'

end

* Em lua é o ciclo If é sempre composto pelo ‘then’ e termina sempre com ‘end’

**if** <condição> **then**  
  
<instruções>  
  
**end**

Problema: verificar se um número introduzido pelo utilizador é par ou ímpar

Nota: Usar condição if else.

Inicio

n

N%2=0

false

true

“Par”

“Impar”

Fim

**Em Python ciclo If/else**

n= input('Digite um numero: ')

if int(n)%2==0:

print('Numero Par')

else:

print('Numero Impar')

**if** <condição>**:**

<instruções>

**else:**

<instruções>

**Em Lua ciclo If/else**

**if** <condição> **then**  
  
<instruções>  
  
**else**  
  
<instruções>  
  
**end**

io.write'Digite um número:'

n = io.read()

if n%2==0 then

print 'Numero Par'

else

print 'Numero Impar'

end

Problema: Escrever um n.º de 1 a 10

Nota: Usar a condição while

I=i + 1

i

I=1

Inicio

I <= 10

true

false

Fim

**Em Python ciclo While**

i=1

while int(i)<=10:

print(i)

i=i+1

* Não existe o ‘end’ para terminar o ciclo, o compilador sabe o que está dentro e fora do ciclo pela maneira como o código está indentado. É sempre necessário o uso dos ‘:’ antes do bloco de comandos

**while** <condição>**:**

   <instruções>

**Em Lua ciclo While**

i=1

while i<=10 do

print(i)

i=i+1

end

* Em Lua o ciclo While é composto sempre pelo ‘do’, o ciclo é terminado com um ‘end’.

**while** <condição> **do**  
  
<instruções>  
  
**end**

Problema: Pedir um número positivo

Nota: usar condição *do while*

I

I < 0

Inicio

I = 0

true

false

Fim

Problema: Factorial de um numero

Nota: uso de funções. Chamada da função fact()

Inicio

I = 5

K \* fact(k - 1)

Fim

k

k > 2

Fact ( k )

J

J = fact ( i )

false True

Fim

**Funções em Python**

**def**<nome\_função>:

<instruções>

k=input('Digite um numero: ')

def fact(k):

if k == 0:

return 1

else:

return (int(k) \* fact( int(k) - 1 ))

j=fact(k)

print(fact(k))

**Funções em Lua**

io.write'Digite um numero: '

**Function**<nome\_função>

<Instruções>

**end**

k=io.read()

function fact(k)

if tonumber(k) == 0 then

return 1

else

return k \* fact(k - 1)

end

end

j=fact(k)

print(j)

**Inicio e Fim**

* Não existe estas equivalências em Python.
* Não existe estas equivalências em Lua.

**Recepção de variáveis em Python**

<variável>= input('<texto>')

**Recepção de variáveis em Lua**

io.write'<texto> '

variavel = io.read()

**Escrever em Python**

Print '<texto>'

**Escrever em Lua**

Print '<texto>'

**Operadores em Python:**

Maior ( > )

Menor ( < )

Maior ou igual ( >= )

Menor ou igual ( <= )

Diferente ( != )

Igual ( = )

E (and )

Ou ( or )

Soma ( + )

Subtracção ( - )

Multiplicação ( \* )

Divisão ( / )

Potenciação ( \*\* ) por exemplo 2\*\*3(2 elevado a 3)

Resto da divisão ( % )

**Operadores em Lua:**

Maior ( > )

Menor ( < )

Maior ou igual ( >= )

Menor ou igual ( <= )

Diferente ( ~= )

Igual ( = )

E (and )

Ou ( or )

Soma ( + )

Subtracção ( - )

Multiplicação ( \* )

Divisão ( / )

Potenciação ( ^ ) por exemplo 2^3(2 elevado a 3)

Resto da divisão ( % )