

# Python

## Lição 1 - Ordem Decrescente

```
In [1]: n1 = int(input ('Digite um valor: '))
s = n1 + 1
m = n1
d = n1 - 1
print ('{} {} {}'.format(s, m, d))
```

6 5 4

## Lição 2 - Dobro, Triplo e Divisão por ela mesma

```
In [2]: n1 = int(input ('Digite um valor: '))
s = n1 + n1
m = n1 + n1 + n1
d = n1 / n1
print ('{} {} {}'.format(s, m, d))
```

10 15 1.0

## Lição 3 - Soma e Média dos Números

```
In [3]: n1 = float(input ('Digite um valor: '))
n2 = float(input ('Digite outro valor: '))
s = n1 + n2
m = (n1 + n2) / 2
print ('{:1f} {:1f}'.format(s, m))
```

12.0 6.0

## Lição 4 - m, dm, cm, mm, hm, dam e km

```
In [1]: n1 = float(input ('Digite um valor: '))
s = n1
g = n1 * 10
m = n1 * 100
d = n1 * 1000
h = n1 / 100
q = n1 / 10
y = n1 / 1000
print ('{}m {}dm {}cm {}mm {}hm {}dam {}km'.format(s, g, m, d, h, q, y))
```

5.0m 50.0dm 500.0cm 5000.0mm 0.05hm 0.5dam 0.005km

## Lição 5 - Tabuada de Multiplicação

```
In [2]: n1 = int(input ('Digite um valor: '))
a = n1 * 1
b = n1 * 2
```

```
c = n1 * 3
d = n1 * 4
e = n1 * 5
f = n1 * 6
g = n1 * 7
h = n1 * 8
i = n1 * 9
j = n1 * 10
print ('{} {} {} {} {} {} {} {} {}'.format(a, b, c, d, e, f, g, h, i))
```

7 14 21 28 35 42 49 56 63 70

## Lição 6 - Conversor de Moedas

```
In [3]: n1 = float(input ('Digite um valor: '))
s = n1
m = n1 * 553550.54
d = n1 * 97121.36
e = n1 * 93090.18
print ('฿{:.2f} = R${:.2f} = ${:.2f} = €{:.2f}'.format(s, m, d, e))
```

฿1.00 = R\$553550.54 = \$97121.36 = €93090.18

## Lição 7 - Altura, Largura, Metro Quadrado e Litros

```
In [4]: n1 = int(input ('Digite um valor: '))
n2 = int(input ('Digite outro valor: '))
s = n1
m = n2
d = n1 * n2
y = (n1 * n2) / 2
print ('LA{}m AL{}m {}m² {}L'.format(s, m, d, y))
```

LA4m AL6m 24m<sup>2</sup> 12.0L

## Lição 8 - Imposto

```
In [5]: n1 = int(input ('Digite um valor: '))
s = n1
m = n1 * 5 / 100
d = n1 - m
print ('R${} -R${} = R${}'.format(s, m, d))
```

R\$1200 -R\$60.0 = R\$1140.0

## Lição 9 - Aumento de Salário

```
In [6]: n1 = int(input ('Digite um valor: '))
s = n1
m = n1 / 2 / 10 / 2 * 6
d = n1 + m
print ('R${} R${} R${}'.format(s, m, d))
```

R\$800 R\$120.0 R\$920.0

## Lição 10 - Dobro, Triplo e Raíz Quadrada

```
In [8]: n1 = int(input ('Digite um valor: '))
s = n1 + n1
m = n1 + n1 + n1
d = n1 ** (1/2)
print ('{} {} {}'.format(s, m, d))
```

288 432 12.0

## Lição 11 - Celsius e Fanherit

```
In [2]: n1 = int(input ('Digite um valor: '))
s = n1
m = ((9 * s) / 5) + 32
print ('{}°C {}°F'.format(s, m))
```

32°C 89.6°F

## Lição 12 - Aluguel de Carros

```
In [3]: dias = int(input ('Quantos dias alugados? '))
km = float(input ('Quantos km rodados? '))
pago = (dias * 60) + (km * 0.15)
print ('o total a pagar é de R${:.2f}'.format(pago))
```

o total a pagar é de R\$182.25

## Lição 13 - Jogos de Azar

```
In [4]: import random
num = random.randint(1, 90)
print(num)
```

78

## Lição 14 - Converter Números Naturais Para Números Inteiros

```
In [5]: '''from math import trunc
num = float(input('Digite um valor: '))
print('O valor digitado foi {} e a sua porção inteira é {}'.format(num,
    num = float(input('Digite um valor: '))
    print('O valor digitado foi {} e a sua porção inteira é {}'.format(num,
```

O valor digitado foi 7.25781 e a sua porção inteira é 7

## Lição 15 - O Cateto e a Hipotenusa

```
In [6]: from math import hypot
co = float(input('Comprimento do cateto oposto: '))
```

```
ca = float(input('Comprimento do cateto adjacente: '))
hi = hypot(co, ca)
print('A hipotenusa vai medir {:.2f}'.format(hi))
```

A hipotenusa vai medir 8.54

## Lição 16 - Como usar e instalar Emojis?

```
In [7]: print('\U0001F468')
print('\U0001F7E6')
print('\U0001F468')
print('\U0001F469')
print('\U0001F409')
print('\U0001F428')
print('\U0001F399')
print('\U0001F4E9')
print('\U0001F159')
print('\U0001F751')
print('\U0001F347')
print('\U0001F178')
print('\U0001f499')
```



Dica: um emoji precisa ser instalado para funcionar e não vem instalado no pacote Python ([É recomendável instalar o Emoji 1.6.1 Emoji for Python](#))

## Lição 17 - Seno, Cosseno e Tangente

```
In [8]: from math import radians, sin, cos, tan
ângulo = float(input('Digite o ângulo que você deseja: '))
seno = sin(radians(ângulo))
print('O ângulo de {} tem o SENO de {:.2f}'.format(ângulo, seno))
cosseno = cos(radians(ângulo))
print('O ângulo de {} tem o COSSENO de {:.2f}'.format(ângulo, cosseno))
tangente = tan(radians(ângulo))
print('O ângulo de {} tem a TANGENTE de {:.2f}'.format(ângulo, tangente))
```

O ângulo de 45.0 tem o SENO de 0.71  
 O ângulo de 45.0 tem o COSSENO de 0.71  
 O ângulo de 45.0 tem a TANGENTE de 1.00

## Lição 18 - Sorteio de Alunos

```
In [9]: from random import choice
n1 = str(input('Primeiro aluno: '))
n2 = str(input('Segundo aluno: '))
n3 = str(input('Terceiro aluno: '))
n4 = str(input('Quarto aluno: '))
lista = [n1, n2, n3, n4]
escolhido = choice(lista)
print('O aluno escolhido foi {}'.format(escolhido))
```

O aluno escolhido foi Maria

## Lição 19 - Sorteio de Ordem de Apresentação

```
In [10]: from random import shuffle
n1 = str(input('Primeiro aluno: '))
n2 = str(input('Segundo aluno: '))
n3 = str(input('Terceiro aluno: '))
n4 = str(input('Quarto aluno: '))
lista = [n1, n2, n3, n4]
shuffle(lista)
print('A ordem de apresentação será ')
print(lista)
```

A ordem de apresentação será  
['Vitor', 'Luísa', 'Ana', 'Lucas']

## Lição 20 - Executar e Criar Arquivo .mp3

```
In [11]: import playsound
playsound.playsound('floatinghomebrianbolger.mp3')
```

```
-----
-
ModuleNotFoundError                                     Traceback (most recent call last)
st)
Cell In[11], line 1
----> 1 import playsound
      2 playsound.playsound('floatinghomebrianbolger.mp3')

ModuleNotFoundError: No module named 'playsound'
```

**pip install playsound** No Prompt de Comando (.cmd)

## Lição 21 - Analizando as Letras

```
In [12]: nome = str(input('Digite seu nome completo: ')).strip()
print('Analizando seu nome...')
print('Seu nome em maiúsculas é {}'.format(nome.upper()))
print('Seu nome em minúsculas é {}'.format(nome.lower()))
print('Seu nome tem ao todo {} letras'.format(len(nome) - nome.count(' ')))
#print('Seu primeiro nome tem {} letras'.format(nome.find(' ')))
separa = nome.split()
print('Seu primeiro nome é {} e ele tem {} letras'.format(separa[0], len(separa[0])))
```

Analizando seu nome...  
 Seu nome em maiúsculas é ENZO DOS SANTOS NEVES  
 Seu nome em minúsculas é enzo dos santos neves  
 Seu nome tem ao todo 18 letras  
 Seu primeiro nome é Enzo e ele tem 4 letras

## Lição 22 - Unidade, Dezena, Centena e Milhar

```
In [13]: num = int(input('Informe um número: '))
u = num // 1 % 10
d = num // 10 % 10
c = num // 100 % 10
m = num // 1000 % 10
print('Analizando o número {}'.format(num))
print('Unidade: {}'.format(u))
print('Dezena: {}'.format(d))
print('Centena: {}'.format(c))
print('Milhar: {}'.format(m))
```

Analizando o número 7865  
 Unidade: 5  
 Dezena: 6  
 Centena: 8  
 Milhar: 7

## Lição 23 - Identificando Falsas Cidades

```
In [14]: cid = str(input('Em que cidade você nasceu? ')).strip()
print(cid[:5].upper() == 'MAUÁ')
```

False

## Lição 24 - Identificando Nomes Falsos

```
In [15]: nome = str(input('Qual é seu nome completo? ')).strip()
print('Seu nome tem Silva? {}'.format('silva' in nome.lower()))
```

Seu nome tem Silva? False

## Lição 25 - Quantas Letras "A" tem em uma Palavra?

```
In [16]: frase = str(input('Digite uma frase: ')).upper().strip()
print ('A letra A aparece {} vezes na frase.'.format(frase.count('A')))
print ('A primeira letra A apareceu na posição {}.'.format(frase.find('A')))
print ('A última letra A apareceu na posição {}.'.format(frase.rfind('A')))
```

A letra A aparece 4 vezes na frase.  
 A primeira letra A apareceu na posição 1.  
 A última letra A apareceu na posição 7.

## Lição 26 - Primeiro e Último Nome

```
In [1]: n = str(input('Digite seu nome completo: ')).strip()
nome = n.split()
print('Muito prazer em te conhecer!')
print('Seu primeiro nome é {}'.format(nome[0]))
print('Seu último nome é {}'.format(nome[len(nome)-1]))
```

Muito prazer em te conhecer!  
 Seu primeiro nome é Enzo  
 Seu último nome é Neves

## Lição 27 - Jogo da Adivinhação

```
In [4]: from random import randint
from time import sleep
computador = randint(0, 5) # Faz o computador "PENSAR"
print('=-' * 20)
print('Vou pensar em um número entre 0 e 5. Tente adivinhar...')
print('=-' * 20)
jogador = int(input('Em que número eu pensei? '))
print('PROCESSANDO... ')
sleep(3)
if jogador == computador:
    print('PARABÉNS! Você conseguiu me vencer!')
else:
    print('GANHEI! Eu pensei no número {} e não no {}!'.format(computador, jogador))
```

Vou pensar em um número entre 0 e 5. Tente adivinhar...  
 PROCESSANDO...  
 PARABÉNS! Você conseguiu me vencer!

## Lição 28 - Radar de Velocidade

```
In [3]: velocidade = float(input('Qual é a velocidade atual do carro? '))
if velocidade > 80:
    print('MULTADO! Você excedeu o limite permitido que é de 80km/h')
    multa = (velocidade-80) * 7
    print('Você deve pagar uma multa de R${:.2f}'.format(multa))
print('Tenha um bom dia! Dirija com segurança!')
```

MULTADO! Você excedeu o limite permitido que é de 80km/h  
 Você deve pagar uma multa de R\$70.00  
 Tenha um bom dia! Dirija com segurança!

## Lição 29 - Par ou Ímpar

```
In [5]: número = int(input('Me diga um número qualquer: '))
resultado = número % 2
if resultado == 0:
    print('O número {} é PAR'.format(número))
else:
    print('O número {} é ÍMPAR'.format(número))
```

O número 4 é PAR

## Lição 30 - Custo de Viagem

```
In [6]: distância = float(input('Qual é a distância da sua viagem? '))
print('Você está prestes a começar uma viagem de {}km.'.format(distância))
preço = distância * 0.50 if distância <= 200 else distância * 0.45
print('E o preço da sua passagem será de R${:.2f}'.format(preço))
```

Você está prestes a começar uma viagem de 1971.0km.  
E o preço da sua passagem será de R\$886.95

## Lição 31 - Ano Bиссexto

```
In [7]: from datetime import date
ano = int(input('Que ano quer analisar? Coloque 0 para analisar o ano atual'))
if ano == 0:
    ano = date.today().year
if ano % 4 == 0 and ano % 100 != 0 or ano % 400 == 0:
    print('O ano {} é BISSEXTO'.format(ano))
else:
    print('O ano {} NÃO é BISSEXTO'.format(ano))
```

O ano 2025 NÃO é BISSEXTO

## Lição 32 - Maior e Menor Número

```
In [8]: a = int(input('Primeiro valor: '))
b = int(input('Segundo valor: '))
c = int(input('Terceiro valor: '))
# Verificando quem é menor
menor = a
if b < a and b < c:
    menor = b
if c < a and c < b:
    menor = c
# Verificando quem é maior
maior = a
if b > a and b > c:
    maior = b
if c > a and c > b:
    maior = c
print('O menor valor digitado foi {}'.format(menor))
print('O maior valor digitado foi {}'.format(maior))
```

O menor valor digitado foi 3  
O maior valor digitado foi 8

## Lição 33 - Reajuste Salarial

```
In [9]: salário = float(input('Qual é o salário do funcionário? R$'))
if salário <= 1250:
    novo = salário + (salário * 15 / 100)
else:
    novo = salário + (salário * 10 / 100)
print('Quem ganhava R${:.2f} passa a ganhar R${:.2f} agora.'.format(salário, novo))
```

Quem ganhava R\$150000.00 passa a ganhar R\$165000.00 agora.

## Lição 34 - Analisador de Triângulos

```
In [10]: print('=-*20')
print('Analizador de triângulos')
print('=-*20')
r1 = float(input('Primeiro segmento: '))
r2 = float(input('Segundo segmento: '))
r3 = float(input('Terceiro segmento: '))
if r1 < r2 + r3 and r2 < r1 + r3 and r3 < r1 + r2:
    print('Os segmentos acima PODEM FORMAR triângulo!')
else:
    print('Os segmentos acima NÃO PODEM FORMAR triângulo')

=====
Analizador de triângulos
=====
Os segmentos acima PODEM FORMAR triângulo!
```

## Lição 35 - (if...elif), (for) e (while)

```
In [11]: nome = str(input('Qual é o seu nome? '))
if nome == 'Gustavo':
    print('Que nome bonito!')
elif nome == 'Pedro' or nome == 'Maria' or nome == 'Paulo':
    print('Seu nome é bem popular no Brasil.')
elif nome in 'Ana Cláudia Jéssica Juliana':
    print('Belo nome feminino!')
else:
    print('Seu nome é bem normal.')
print('Tenha um bom dia, {}'.format(nome))
```

Seu nome é bem normal.  
Tenha um bom dia, Paula!

## Lição 36 - Aprovando Empréstimo

```
In [12]: casa = float(input('Valor da casa: R$'))
salário = float(input('Salário do comprador: R$'))
anos = int(input('Quantos anos de financiamento? '))
prestação = casa / (anos * 12)
mínimo = salário * 30 / 100
print('Para pagar uma casa de R${:.2f} em {} anos'.format(casa, anos), end=' ')
print(' a prestação será de R${:.2f}'.format(prestação))
if prestação <= mínimo:
    print('Empréstimo deve ser CONCEDIDO!')
else:
    print('Empréstimo NEGADO!')
```

Para pagar uma casa de R\$120000.00 em 10 anos a prestação será de R\$1000.00  
Empréstimo deve ser CONCEDIDO!

## Lição 37 - Conversor de Bases Numéricas

```
In [13]: num = int(input('Digite um número inteiro: '))
print('''Escolha uma das bases para conversão:
[ 1 ] converter para BINÁRIO
[ 2 ] converter para OCTAL
[ 3 ] converter para HEXADECIMAL''')
opção = int(input('Sua opção: '))
if opção == 1:
    print('{} convertido para BINÁRIO é igual a {}'.format(num, bin(num)))
elif opção == 2:
    print('{} convertido para OCTAL é igual a {}'.format(num, oct(num)))
elif opção == 3:
    print('{} convertido para HEXADECIMAL é igual a {}'.format(num, hex(num)))
else:
    print('Opção inválida. tente novamente.')
```

Escolha uma das bases para conversão:  
[ 1 ] converter para BINÁRIO  
[ 2 ] converter para OCTAL  
[ 3 ] converter para HEXADECIMAL  
250 convertido para BINÁRIO é igual a 0b11111010

## Lição 38 - Comparando Números

```
In [14]: n1 = (int(input('Primeiro número: ')))
n2 = (int(input('Segundo número: ')))
if n1 > n2:
    print('O PRIMEIRO valor é maior')
elif n2 > n1:
    print('O SEGUNDO valor é maior')
else:
    print('Os dois valores são IGUAIS')
```

Os dois valores são IGUAIS

## Lição 39 - Alistamento Militar

```
In [15]: from datetime import date
ano_nascimento = int(input('Em que ano você nasceu? '))
sexo = str(input('Você é HOMEM ou MULHER? ')).title().strip()
idade = date.today().year - ano_nascimento
if sexo == 'Mulher':
    print('Você não é obrigada ao alistamento.')
    sim_nao = str(input('Você Gostaria de se alistar nas Forças Armadas? S: '))
    if sim_nao == 'Não':
        print('Tenha um bom dia Senhorita!')
    elif sim_nao == 'Sim':
        print('BEM-VINDA Ao Serviço Militar!')
        if idade == 18:
            print('Você tem que se alistar IMEDIATAMENTE!')
        elif idade > 18:
            alistamento = idade - 18
            ano = date.today().year - alistamento
            print('Você já deveria ter se alistado há {} ano(s)'.format(alistamento))
            print('Seu alistamento foi em {}'.format(ano))
        else:
            alistamento = 18 - idade
            ano = alistamento + date.today().year
    else:
        print('Opção inválida. tente novamente.')
```

```

        print('Ainda faltam {} ano(s) para o alistamento'.format(alistamento)
        print('Seu alistamento será em {}'.format(ano))
    else:
        print('Entrada inválida, escolher Sim ou Não')
    elif sexo == 'Homem':
        print('BEM-VINDO ao Serviço Militar')
        if idade == 18:
            alistamento = 18
            print('Você tem que se alistar IMEDIATAMENTE!')
        elif idade > 18:
            alistamento = idade - 18
            ano = date.today().year - alistamento
            print('Você já deveria ter se alistado há {} ano(s)'.format(alistamento))
            print('Seu alistamento foi em {}'.format(ano))
        else:
            alistamento = 18 - idade
            ano = alistamento + date.today().year
            print('Ainda faltam {} ano(s) para o alistamento'.format(alistamento))
            print('Seu alistamento será em {}'.format(ano))
    else:
        print('As Forças Armadas não aceita DEGENERADO(A)!')

```

BEM-VINDO ao Serviço Militar  
 Você já deveria ter se alistado há 2 ano(s)  
 Seu alistamento foi em 2023

## Lição 40 - Média Escolar

```

In [16]: nota1 = float(input('Primeira nota: '))
nota2 = float(input('Segunda nota: '))
média = (nota1 + nota2) / 2
print('Tirando {:.1f} e {:.1f}, a média do aluno é {:.1f}'.format(nota1
if 7 > média >=5:
    print('O aluno está de RECUPERAÇÃO.')
elif média < 5:
    print('O aluno está REPROVADO.')
elif média >= 7:
    print('O aluno está APROVADO.')

```

Tirando 2.5 e 9.5, a média do aluno é 6.0  
 O aluno está de RECUPERAÇÃO.

## Lição 41 - Classificando Atletas

```

In [17]: from datetime import date
atual = date.today().year
nascimento = int(input('Ano de Nascimento: '))
idade = atual - nascimento
print('O atleta tem {} anos'.format(idade))
if idade <= 9:
    print('Classificação: MIRIM')
elif idade <= 14:
    print('Classificação: INFANTIL')
elif idade <= 19:
    print('Classificação: JUNIOR')
elif idade <= 25:
    print('Classificação: SÊNIOR')

```

```
else:
    print('Classificação: MASTER')
```

O atleta tem 20 anos  
Classificação: SÊNIOR

## Lição 42 - Analisador de Triângulos 2.0

```
In [18]: r1 = float(input('Primeiro segmento: '))
r2 = float(input('Segundo segmento: '))
r3 = float(input('Terceiro segmento: '))
if r1 < r2 + r3 and r2 < r1 + r3 and r3 < r1 + r2:
    print('Os segmentos acima PODEM FORMAR triângulo ', end=' ')
    if r1 == r2 == r3:
        print('EQUILÁTERO!')
    elif r1 != r2 != r3 != r1:
        print('ESCALENO!')
    else:
        print('ISÓCELES')
else:
    print('Os segmentos acima NÃO PODEM FORMAR triângulo!')
```

Os segmentos acima PODEM FORMAR triângulo EQUILÁTERO!

## Lição 43 - Índice de Massa Corporal

```
In [19]: peso = float(input('Qual é seu peso? (kg) '))
altura = float(input('Qual é sua altura? (m) '))
imc = peso / (altura ** 2)
print('O IMC dessa pessoa é de {:.1f}'.format(imc))
if imc < 18.5:
    print('Você está ABAIXO DO PESO normal')
elif 18.5 <= imc < 25:
    print('PARABÉNS, você está na faixa do PESO NORMAL')
elif 25 <= imc < 30:
    print('Você está em SOBREPESO')
elif 30 <= imc < 40:
    print('Você está em OBESIDADE')
elif imc >= 40:
    print('Você está em OBESIDADE MÓRBIDA, cuidado!')
```

O IMC dessa pessoa é de 20.3  
PARABÉNS, você está na faixa do PESO NORMAL

## Lição 44 - Tuplas

```
In [20]: pessoa = ('Gustavo', 39, 'M', 99.88)
print(pessoa)
```

('Gustavo', 39, 'M', 99.88)

Esse é apenas um exemplo de tupla

## Lição 45 - Número por Extenso

```
In [21]: cont = ('zero', 'um', 'dois', 'três', 'quatro',
             'cinco', 'seis', 'sete', 'oito', 'nove',
             'dez', 'onze', 'doze', 'treze', 'catorze',
             'quinze', 'dezesseis', 'dezessete', 'dezoito',
             'dezenove', 'vinte')
while True:
    num = int(input('Digite um número entre 0 e 20: '))
    if 0 <= num <= 20:
        break
    print('Tente novamente. ', end='')
print(f'Você digitou o número {cont[num]}')
```

Você digitou o número oito

## Lição 46 - Tuplas com Times de Futebol

```
In [22]: times = ('Corintians', 'Palmeiras', 'Santos', 'Grêmio',
              'Cruzeiro', 'Flamengo', 'Vasco', 'Chapecoense',
              'Atlético', 'Botafogo', 'Atlético-PR', 'Bahia',
              'São Paulo', 'Fluminense', 'Sport Recife',
              'EC Vitória', 'Coritiba', 'Avaí', 'Ponte Preta',
              'Atlético-GO')
print('=' * 15)
print(f'Lista de times do Brasileirão: {times}')
print('=' * 15)
print(f'Os 5 primeiros são {times[0:5]}')
print('=' * 15)
print(f'Os 4 últimos são {times[-4:]}')
print('=' * 15)
print(f'Times em ordem alfabética: {sorted(times)})')
print('=' * 15)
print(f'O Chapecoense está na {times.index("Chapecoense") + 1}ª posição')
```

=====
Lista de times do Brasileirão: ('Corintians', 'Palmeiras', 'Santos', 'Grêmio', 'Cruzeiro', 'Flamengo', 'Vasco', 'Chapecoense', 'Atlético', 'Botafogo', 'Atlético-PR', 'Bahia', 'São Paulo', 'Fluminense', 'Sport Recife', 'EC Vitória', 'Coritiba', 'Avaí', 'Ponte Preta', 'Atlético-GO')
=====

Os 5 primeiros são ('Corintians', 'Palmeiras', 'Santos', 'Grêmio', 'Cruzeiro')

=====
Os 4 últimos são ('Coritiba', 'Avaí', 'Ponte Preta', 'Atlético-GO')

=====
Times em ordem alfabética: ['Atlético', 'Atlético-GO', 'Atlético-PR', 'Avaí', 'Bahia', 'Botafogo', 'Chapecoense', 'Corintians', 'Coritiba', 'Cruzeiro', 'EC Vitória', 'Flamengo', 'Fluminense', 'Grêmio', 'Palmeiras', 'Ponte Preta', 'Santos', 'Sport Recife', 'São Paulo', 'Vasco']

=====
0 Chapecoense está na 8ª posição

## Lição 47 - Maior e Menor Valores em Tupla

```
In [23]: from random import randint
numeros = (randint(1, 10), randint(1, 10), randint(1, 10),
            randint(1, 10), randint(1, 10))
print('Os valores sorteados foram: ', end='')
```

```

for n in numeros:
    print(f'{n} ', end='')
print(f'\nO maior valor sorteado foi {max(numeros)}')
print(f'O menor valor sorteado foi {min(numeros)}')

```

Os valores sorteados foram: 7 1 6 10 9  
 O maior valor sorteado foi 10  
 O menor valor sorteado foi 1

## Lição 48 - Análise de Dados em uma Tupla

```

In [24]: num = (int(input('Digite um número: ')),
            int(input('Digite outro número: ')),
            int(input('Digite mais um número: ')),
            int(input('Digite o último número: ')))
print(f'Você digitou os valores {num}')
print(f'O valor 9 apareceu {num.count(9)} vezes')
if 3 in num:
    print(f'O valor 3 apareceu na {num.index(3)+1}ª posição')
else:
    print('O valor 3 não foi digitado em nenhuma posição')
print('Os valores pares digitados foram ', end=' ')
for n in num:
    if n % 2 == 0:
        print(n, end=' ')

```

Você digitou os valores (4, 8, 2, 6)  
 O valor 9 apareceu 0 vezes  
 O valor 3 não foi digitado em nenhuma posição  
 Os valores pares digitados foram 4 8 2 6

## Lição 49 - Lista de Preços com Tupla

```

In [25]: listagem = ('Lápis', 1.75,
                  'Borracha', 2,
                  'Caderno', 15.90,
                  'Estojo', 25,
                  'Transferidor', 4.20,
                  'Compasso', 9.99,
                  'Mochila', 120.32,
                  'Canetas', 22.30,
                  'Livro', 34.90)
print('-' * 40)
print(f'{"LISTAGEM DE PREÇOS":^40}')
print('-' * 40)
for pos in range(0, len(listagem)):
    if pos % 2 == 0:
        print(f'{listagem[pos]}:<30}', end=' ')
    else:
        print(f'R${listagem[pos]}:>7.2f')
print('-' * 40)

```

```
-----  
                         LISTAGEM DE PREÇOS  
-----  
Lápis.....R$ 1.75  
Borracha.....R$ 2.00  
Caderno.....R$ 15.90  
Estojo.....R$ 25.00  
Transferidor.....R$ 4.20  
Compasso.....R$ 9.99  
Mochila.....R$ 120.32  
Canetas.....R$ 22.30  
Livro.....R$ 34.90  
-----
```

## Lição 50 - Contando Vogais em Tupla

```
In [26]: palavras = ('aprender', 'programar', 'linguagem', 'python',
               'curso', 'grátis', 'estudar', 'praticar',
               'trabalhar', 'mercado', 'programador', 'futuro')
for p in palavras:
    print(f'\nNa palavra {p.upper()} temos ', end=' ')
    for letra in p:
        if letra.lower() in 'aeiou':
            print(letra, end=' ')
```

Na palavra APRENDER temos a e e  
 Na palavra PROGRAMAR temos o a a  
 Na palavra LINGUAGEM temos i u a e  
 Na palavra PYTHON temos o  
 Na palavra CURSO temos u o  
 Na palavra GRÁTIS temos i  
 Na palavra ESTUDAR temos e u a  
 Na palavra PRATICAR temos a i a  
 Na palavra TRABALHAR temos a a a  
 Na palavra MERCADO temos e a o  
 Na palavra PROGRAMADOR temos o a a o  
 Na palavra FUTURO temos u u o

## Lição 51 - Listas (Parte 1)

```
In [27]: a = [2, 3, 4, 7]
b = a[:]
b[2] = 8
print(f'Lista A: {a}')
print(f'Lista B: {b}')
```

Lista A: [2, 3, 4, 7]  
 Lista B: [2, 3, 8, 7]

Esse é apenas um exemplo de lista

## Lição 52 - Maior e Menor Valores da Lista

```
In [28]: listanum = []
mai = 0
men = 0
```

```

for c in range(0, 5):
    listanum.append(int(input(f'Digite um valor para a Posição {c}: ')))
    if c == 0:
        mai = men = listanum[c]
    else:
        if listanum[c] > mai:
            mai = listanum[c]
        if listanum[c] < men:
            men = listanum[c]
print('=-' * 30)
print(f'Você digitou os valores {listanum}')
print(f'O maior número digitado foi {mai} nas posições ', end=' ')
for i, v in enumerate(listanum):
    if v == mai:
        print(f'{i}... ', end=' ')
print()
print(f'O menor número digitado foi {men} nas posições ', end=' ')
for i, v in enumerate(listanum):
    if v == men:
        print(f'{i}... ', end=' ')
print()

```

=====

Você digitou os valores [4, 8, 6, 5, 3]  
O maior número digitado foi 8 nas posições 1...  
O menor número digitado foi 3 nas posições 4...

## Lição 53 - Valores Únicos em uma Lista

In [29]:

```

números = list()
while True:
    n = int(input('Digite um valor: '))
    if n not in números:
        números.append(n)
        print('Valor adicionado com sucesso...')
    else:
        print('Valor duplicado! não vou adicionar...')
    r = str(input('Quer continuar? [S/N] '))
    if r in 'Nn':
        break
print('=-' * 30)
números.sort()
print(f'Você digitou os valores {números}')

```

Valor adicionado com sucesso...  
Valor adicionado com sucesso...  
Valor duplicado! não vou adicionar...  
Valor adicionado com sucesso...  
=====

Você digitou os valores [6, 7, 8]

## Lição 54 - Lista Ordenada sem Repetições

In [30]:

```

lista = []
for c in range(0, 5):
    n = int(input('Digite um valor: '))
    if c == 0 or n > lista[-1]:
        lista.append(n)

```

```

        print('Adicionando ao final da lista... ')
else:
    pos = 0
    while pos < len(lista):
        if n <= lista[pos]:
            lista.insert(pos, n)
            print(f'Adicionando na posição {pos} da lista...')
            break
        pos += 1
print('=' * 30)
print(f'Os valores digitados em ordem foram {lista}')

```

```

Adicionando ao final da lista...
Adicionando ao final da lista...
Adicionando ao final da lista...
Adicionando na posição 0 da lista...
Adicionando na posição 0 da lista...
=====
Os valores digitados em ordem foram [2, 2, 2, 8, 9]

```

## Lição 55 - Extrair Dados de uma Lista

```

In [31]: valores = []
while True:
    valores.append(int(input('Digite um valor: ')))
    resp = str(input('Quer continuar? [S/N] '))
    if resp in 'Nn':
        break
print('=' * 30)
print(f'Você digitou {len(valores)} elementos.')
valores.sort(reverse=True)
print(f'Os valores em ordem decrescente são {valores}')
if 5 in valores:
    print('O valor 5 faz parte da lista!')
else:
    print('O valor 5 não foi encontrado na lista!')

```

```

=====
Você digitou 4 elementos.
Os valores em ordem decrescente são [9, 7, 5, 3]
O valor 5 faz parte da lista!

```

## Lição 56 - Dividindo Valor em Várias Listas

```

In [32]: num = list()
pares = list()
ímpares = list()
while True:
    num.append(int(input('Digite um número: ')))
    resp = str(input('Quer continuar? [S/N] '))
    if resp in 'Nn':
        break
for i, v in enumerate(num):
    if v % 2 == 0:
        pares.append(v)
    elif v % 2 == 1:
        ímpares.append(v)
print('=' * 30)

```

```
print(f'A lista completa é {num}')
print(f'A lista de pares é {pares}')
print(f'A lista de ímpares é {ímpares}')
```

```
=====
A lista completa é [4, 5]
A lista de pares é [4]
A lista de ímpares é [5]
```

## Lição 57 - Validando Expressões Matemáticas

```
In [33]: expr = str(input('Digite a expressão: '))
pilha = []
for simb in expr:
    if simb == '(':
        pilha.append('(')
    elif simb == ')':
        if len(pilha) > 0:
            pilha.pop()
        else:
            pilha.append(')')
            break
if len(pilha) == 0:
    print('Sua expressão está valida!')
else:
    print('Sua expressão está errada!')
```

Sua expressão está valida!

## Lição 58 - Listas (Parte 2)

```
In [34]: galera = list()
dados = list()
totmai = totmen = 0
for c in range(0, 3):
    dados.append(str(input('Nome: ')))
    dados.append(int(input('Idade: ')))
    galera.append(dados[:])
    dados.clear()

for p in galera:
    if p[1] >= 21:
        print(f'{p[0]} é maior de idade.')
        totmai += 1
    else:
        print(f'{p[0]} é menor de idade.')
        totmen += 1

print(f'Temos {totmai} maiores e {totmen} menores de idade.)
```

Geovana é maior de idade.  
 Nicolly é menor de idade.  
 Kelly é maior de idade.  
 Temos 2 maiores e 1 menores de idade.

Esse é mais um dos exemplos de listas

## Lição 59 - Lista Composta e Análise de Dados

In [36]:

```
temp = []
princ = []
mai = men = 0
while True:
    temp.append(str(input('Nome: ')))
    temp.append(float(input('Peso: ')))
    if len(princ) == 0:
        mai = men = temp[1]
    else:
        if temp[1] > mai:
            mai = temp[1]
        if temp[1] < men:
            men = temp[1]
    princ.append(temp[:])
    temp.clear()
    resp = str(input('Quer continuar? [S/N] '))
    if resp in 'Nn':
        break
print('-=' * 30)
print(f'Ao todo, você cadastrou {len(princ)} pessoas.')
print(f'O maior peso foi de {mai}Kg. Peso de ', end=' ')
for p in princ:
    if p[1] == mai:
        print(f'[{p[0]}] ', end=' ')
print()
print(f'O menor peso foi de {men}Kg. Peso de ', end=' ')
for p in princ:
    if p[1] == men:
        print(f'[{p[0]}] ', end=' ')
print()
```

=====
Ao todo, você cadastrou 3 pessoas.  
O maior peso foi de 100.0Kg. Peso de [Gustavo]  
O menor peso foi de 70.0Kg. Peso de [Curiosildo]

## Lição 60 - Listas com Pares e ímpares

In [37]:

```
núm = [[], []]
valor = 0
for c in range(1, 8):
    valor = int(input(f'Digite o {c}º valor: '))
    if valor % 2 == 0:
        núm[0].append(valor)
    else:
        núm[1].append(valor)
print('-=' * 30)
núm[0].sort()
núm[1].sort()
print(f'Os valores pares digitados foram: {núm[0]}')
print(f'Os valores ímpares digitados foram: {núm[1]}')
```

=====
Os valores pares digitados foram: [2, 4, 6, 8]  
Os valores ímpares digitados foram: [1, 5, 7]

## Lição 61 - Matriz em Python

```
In [38]: matriz = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
for l in range(0, 3):
    for c in range(0, 3):
        matriz[l][c] = int(input(f'Digite um valor para [{l}, {c}]: '))
print('=' * 30)
for l in range(0, 3):
    for c in range(0, 3):
        print(f'{matriz[l][c]:^5}', end=' ')
    print()
=====
[ 7 ][ 5 ][ 6 ]
[ 1 ][ 4 ][ 8 ]
[ 9 ][ 2 ][ 6 ]
```

## Lição 62 - Mais Sobre Matriz em Python

```
In [39]: matriz = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
spar = mai = scol = 0
for l in range(0, 3):
    for c in range(0, 3):
        matriz[l][c] = int(input(f'Digite um valor para [{l}, {c}]: '))
print('=' * 30)
for l in range(0, 3):
    for c in range(0, 3):
        print(f'{matriz[l][c]:^5}', end=' ')
        if matriz[l][c] % 2 == 0:
            spar += matriz[l][c]
    print()
print('=' * 30)
print(f'A soma dos valores pares é {spar}')
for l in range(0, 3):
    scol += matriz[l][2]
print(f'A soma dos valores da terceira coluna é {scol}.')
for c in range(0, 3):
    if c == 0:
        mai = matriz[1][c]
    elif matriz[1][c] > mai:
        mai = matriz[1][c]
print(f'O maior valor da segunda linha é {mai}.')
=====
[ 4 ][ 5 ][ 8 ]
[ 1 ][ 9 ][ 3 ]
[ 2 ][ 7 ][ 6 ]
=====
A soma dos valores pares é 20
A soma dos valores da terceira coluna é 17.
O maior valor da segunda linha é 9.
```

## Lição 63 - Palpites Para a Mega Sena

```
In [40]: from random import randint
from time import sleep
```

```

lista = list()
jogos = list()
print('-' * 30)
print('      JOGA NA MEGA SENA      ')
print('-' * 30)
quant = int(input('Quantos jogos você quer que eu sorteie? '))
tot = 1
while tot <= quant:
    cont = 0
    while True:
        num = randint(1, 60)
        if num not in lista:
            lista.append(num)
            cont += 1
        if cont >= 6:
            break
    lista.sort()
    jogos.append(lista[:])
    lista.clear()
    tot += 1
print('=-' * 3, f' SORTEANDO {quant} JOGOS ', '-=' * 3)
for i, l in enumerate(jogos):
    print(f'Jogo {i+1}: {l}')
    sleep(1)
print('=-' * 5, '< BOA SORTE! >', '-=' * 5)

```

```

-----
JOGA NA MEGA SENA
-----
-==== SORTEANDO 8 JOGOS -====-
Jogo 1: [31, 33, 42, 50, 59, 60]
Jogo 2: [16, 17, 19, 42, 51, 53]
Jogo 3: [6, 8, 30, 39, 43, 51]
Jogo 4: [14, 24, 25, 29, 38, 42]
Jogo 5: [6, 9, 25, 38, 48, 53]
Jogo 6: [9, 14, 20, 35, 48, 49]
Jogo 7: [29, 33, 39, 46, 49, 52]
Jogo 8: [8, 24, 26, 31, 42, 56]
----- < BOA SORTE! > -----

```

## Lição 64 - Boletim com Notas Compostas

```

In [41]: ficha = list()
while True:
    nome = str(input('Nome: '))
    nota1 = float(input('Nota 1: '))
    nota2 = float(input('Nota 2: '))
    media = (nota1 + nota2) / 2
    ficha.append([nome, [nota1, nota2], media])
    resp = str(input('Quer continuar? [S/N] '))
    if resp in 'Nn':
        break
print('-' * 30)
print(f'{ "No. " :<4}{ "NOME " :<10}{ "MÉDIA " :>8 }')
print('-' * 26)
for i, a in enumerate(ficha):
    print(f'{i:<4}{a[0]:<10}{a[2]:>8.1f}')
while True:
    print('-' * 35)

```

```

opc = int(input('Mostrar notas de qual aluno? (999 interrompe): '))
if opc == 999:
    print('FINALIZANDO...')
    break
if opc <= len(ficha) - 1:
    print(f'Notas de {ficha[opc][0]} são {ficha[opc][1]}')
print('<<< VOLTE SEMPRE! >>>')

=====

```

No. NOME MÉDIA

	NOME	MÉDIA
0	Ricardo	6.0
1	Jennifer	8.0

Notas de Ricardo são [8.0, 4.0]

FINALIZANDO...

<<< VOLTE SEMPRE! >>>

## Lição 65 - Dicionários em Python

```

In [42]: estado = dict()
brasil = list()
for c in range(0, 3):
    estado['uf'] = str(input('Unidade Federativa: '))
    estado['sigla'] = str(input('Sigla do Estado: '))
    brasil.append(estado.copy())
for e in brasil:
    for v in e.values():
        print(v, end=' ')
    print()

```

AMazonas AM

T0cantins T0

BaHia BH

## Lição 66 - Dicionário em Python

```

In [43]: aluno = dict()
aluno['nome'] = str(input('Nome: '))
aluno['média'] = float(input(f'Média de {aluno["nome"]}: '))
if aluno['média'] >= 7:
    aluno['situação'] = 'Aprovado'
elif 5 <= aluno['média'] < 7:
    aluno['situação'] = 'Recuperação'
else:
    aluno['situação'] = 'Reprovado'
print('-=' * 30)
for k, v in aluno.items():
    print(f' - {k} é igual a {v}')

```

=====

- nome é igual a Fernanda Souza de Moraes
- média é igual a 10.0
- situação é igual a Aprovado

## Lição 67 - Jogo de Dados em Python

```
In [44]: from random import randint
from time import sleep
from operator import itemgetter
jogo = {'jogador1': randint(1, 6),
        'jogador2': randint(1, 6),
        'jogador3': randint(1, 6),
        'jogador4': randint(1, 6)}
ranking = list()
print('Valores Sorteados:')
for k, v in jogo.items():
    print(f'{k} tirou {v} no dado.')
    sleep(1)
ranking = sorted(jogo.items(), key=itemgetter(1), reverse=True)
print('=-' * 30)
print(' == RANKING DOS JOGADORES ==')
for i, v in enumerate(ranking):
    print(f'    {i+1}º lugar: {v[0]} com {v[1]}.')
    sleep(1)
```

Valores Sorteados:  
jogador1 tirou 6 no dado.  
jogador2 tirou 3 no dado.  
jogador3 tirou 3 no dado.  
jogador4 tirou 5 no dado.

=====

== RANKING DOS JOGADORES ==  
1º lugar: jogador1 com 6.  
2º lugar: jogador4 com 5.  
3º lugar: jogador2 com 3.  
4º lugar: jogador3 com 3.

## Lição 68 - Cadastro de Trabalhador em Python

```
In [45]: from datetime import datetime
dados = dict()
dados['nome'] = str(input('Nome: '))
nasc = int(input('Ano de nascimento: '))
dados['idade'] = datetime.now().year - nasc
dados['ctps'] = int(input('Carteira de trabalho (0 não tem): '))
if dados['ctps'] != 0:
    dados['contratação'] = int(input('Ano de contratação: '))
    dados['salário'] = float(input('Salário: R$'))
    dados['aposentadoria'] = dados['idade'] + ((dados['contratação'] + 35) - nasc)
print('=-' * 30)
for k, v in dados.items():
    print(f' - {k} tem o valor {v}'')
```

=====

- nome tem o valor José Vieira dos Santos  
- idade tem o valor 58  
- ctps tem o valor 37  
- contratação tem o valor 1985  
- salário tem o valor 1000.0  
- aposentadoria tem o valor 53

## Lição 69 - Cadastro de Jogador de Futebol

```
In [46]: jogador = dict()
partidas = list()
jogador['nome'] = str(input('Nome do jogador: '))
tot = int(input(f'Quantas partidas {jogador["nome"]} já jogou? '))
for c in range(0, tot):
    partidas.append(int(input(f'    quantos gols na partida {c}? ')))
jogador['gols'] = partidas[:]
jogador['total'] = sum(partidas)
print('=' * 30)
print(jogador)
print('=' * 30)
for k, v in jogador.items():
    print(f'O campo {k} tem o valor {v}')
print('=' * 30)
print(f'O jogador {jogador["nome"]} jogou {len(jogador["gols"])} partidas')
for i, v in enumerate(jogador['gols']):
    print(f'    => Na partida {i}, fez {v} gols.')
print(f'Foi um total de {jogador["total"]} gols.')

=====
{'nome': 'Brunão', 'gols': [3, 2, 0, 4], 'total': 9}
=====
0 campo nome tem o valor Brunão
0 campo gols tem o valor [3, 2, 0, 4]
0 campo total tem o valor 9
=====
0 jogador Brunão jogou 4 partidas.
    => Na partida 0, fez 3 gols.
    => Na partida 1, fez 2 gols.
    => Na partida 2, fez 0 gols.
    => Na partida 3, fez 4 gols.
Foi um total de 9 gols.
```

## Lição 70 - Unindo Dicionários e Listas

```
In [47]: galera = list()
pessoa = dict()
soma = média = 0
while True:
    pessoa.clear()
    pessoa['nome'] = str(input('Nome: '))
    while True:
        pessoa['sexo'] = str(input('Sexo: [M/F] ')).upper()[0]
        if pessoa['sexo'] in 'MF':
            break
        print('ERRO! Por favor, digite apenas M ou F.')
    pessoa['idade'] = int(input('Idade: '))
    soma += pessoa['idade']
    galera.append(pessoa.copy())
    while True:
        resp = str(input('Quer continuar? [S/N] ')).upper()[0]
        if resp in 'SN':
            break
        print('ERRO! Responda apenas S ou N.')
    if resp == 'N':
```

```

        break
print('=' * 30)
print(f'A) Ao todo temos {len(galera)} pessoas cadastradas.')
média = soma / len(galera)
print(f'B) A média de idade é de {média:.2f} anos.')
print('C) As mulheres cadastradas foram ', end='')
for p in galera:
    if p['sexo'] in 'Ff':
        print(f'{p["nome"]}', end=' ')
print()
print('D) Lista das pessoas que estão acima da média: ')
for p in galera:
    if p['idade'] >= média:
        print(' ', end=' ')
        for k, v in p.items():
            print(f'{k} = {v}; ', end=' ')
        print()
print('<< ENCERRADO >>')

```

- =====
- A) Ao todo temos 4 pessoas cadastradas.  
B) A média de idade é de 20.00 anos.  
C) As mulheres cadastradas foram Ana Carolina  
D) Lista das pessoas que estão acima da média:  
 nome = Ana; sexo = F; idade = 22;  
 nome = Carolina; sexo = F; idade = 25;  
<< ENCERRADO >>

## Lição 71 - Aprimorando os Dicionários

```

In [48]: time = list()
jogador = dict()
partidas = list()
while True:
    jogador.clear()
    jogador['nome'] = str(input('Nome do jogador: '))
    tot = int(input(f'Quantas partidas {jogador["nome"]} já jogou? '))
    partidas.clear()
    for c in range(0, tot):
        partidas.append(int(input(f'    quantos gols na partida {c}? ')))
    jogador['gols'] = partidas[:]
    jogador['total'] = sum(partidas)
    time.append(jogador.copy())
    while True:
        resp = str(input('Quer continuar [S/N] ')).upper()[0]
        if resp in 'SN':
            break
        print('ERRO! responda apenas S ou N.')
    if resp == 'N':
        break
print('=' * 30)
print('cod ', end='')
for i in jogador.keys():
    print(f'{i:<15}', end=' ')
print()
print('-' * 40)
for k, v in enumerate(time):
    print(f'{k:>3} ', end=' ')
    for d in v.values():

```

```

        print(f'{str(d):<15}', end=' ')
    print()
print('-' * 40)
while True:
    busca = int(input('Mostrar dados de qual jogador? (999 para parar)')
    if busca == 999:
        break
    if busca >= len(time):
        print(f'ERRO! Não existe jogador com código {busca}!')
    else:
        print(f' -- LEVANTAMENTO DO JOGADOR {time[busca]["nome"]}:')
        for i, g in enumerate(time[busca]['gols']):
            print(f'     No jogo {i+1} fez {g} gols.')
    print('-' * 40)
print('<< VOLTE SEMPRE! >>')

=====
cod nome          gols          total
-----
  0 Pedro          [3, 2, 4, 1]    10
  1 Brunão         [0, 1]           1
-----
-- LEVANTAMENTO DO JOGADOR Pedro:
  No jogo 1 fez 3 gols.
  No jogo 2 fez 2 gols.
  No jogo 3 fez 4 gols.
  No jogo 4 fez 1 gols.
-----
-- LEVANTAMENTO DO JOGADOR Brunão:
  No jogo 1 fez 0 gols.
  No jogo 2 fez 1 gols.
-----
<< VOLTE SEMPRE! >>

```

## Lição 72 - Funções (Parte 1)

```
In [49]: def soma(* valores):
    s = 0
    for num in valores:
        s += num
    print(f'Somando os valores {valores} temos {s}')

soma(5, 2)
soma(2, 9, 4)
```

Somando os valores (5, 2) temos 7  
 Somando os valores (2, 9, 4) temos 15

Esse é um exemplo de função

## Lição 73 - Função que Calcula Área

```
In [50]: def área(larg, comp):
    a = larg * comp
    print(f'A área de um terreno {larg}x{comp} é de {a}m².')
```

```
# Programa Principal
print(' Controle de terrenos')
print('-' * 20)
l = float(input('LARGURA (m): '))
c = float(input('COMPRIMENTO (m): '))
área(l, c)
```

Controle de terrenos

A área de um terreno 2.0x8.0 é de 16.0m<sup>2</sup>.

## Lição 74 - Um Print Especial

In [51]:

```
def escreva(msg):
    tam = len(msg) + 4
    print('~' * tam)
    print(f' {msg} ')
    print('~' * tam)
```

```
# Programa Principal
escreva('Gustavo guanabara')
escreva('Curso de Python no YouTube')
escreva('CeV')
```

~~~~~  
Gustavo guanabara  
~~~~~

~~~~~  
Curso de Python no YouTube  
~~~~~

~~~~~  
CeV  
~~~~~

## Lição 75 - Função de Contador

In [52]:

```
from time import sleep

def contador(i, f, p):
    if p < 0:
        p *= -1
    if p == 0:
        p = 1
    print('=-' * 20)
    print(f'Contagem de {i} até {f} de {p} em {p}')
    sleep(2.5)

    if i < f:
        cont = i
        while cont <= f:
            print(f'{cont} ', end=' ', flush=True)
            sleep(0.5)
            cont += p
        print('FIM!')
    else:
        cont = i
```

```

        while cont >= f:
            print(f'{cont} ', end='', flush=True)
            sleep(0.5)
            cont -= p
        print('FIM!')

#Programa Principal
contador(1, 10, 1)
contador(10, 0, 2)
print('=-' * 20)
print('Agora é sua vez de personalizar a contagem!')
ini = int(input('Início: '))
fim = int(input('Fim: '))
pas = int(input('Passo: '))
contador(ini, fim, pas)

=====
Contagem de 1 até 10 de 1 em 1
1 2 3 4 5 6 7 8 9 10 FIM!
=====
Contagem de 10 até 0 de 2 em 2
10 8 6 4 2 0 FIM!
=====
Agora é sua vez de personalizar a contagem!
=====
Contagem de 5 até 50 de 5 em 5
5 10 15 20 25 30 35 40 45 50 FIM!

```

## Lição 76 - Função que Descobre o Maior

```

In [53]: from time import sleep

def maior(* núm):
    cont = maior = 0
    print('=-' * 30)
    print('Analisando os valores passados... ')
    for valor in núm:
        print(f'{valor} ', end='', flush=True)
        sleep(0.3)
        if cont == 0:
            maior = valor
        else:
            if valor > maior:
                maior = valor
        cont += 1
    print(f'Foram informados {cont} valores ao lado.')
    print(f'O maior valor informado foi {maior}.')

# Programa Principal
maior(2, 9, 4, 5, 7, 1)
maior(4, 7, 0)
maior(1, 2)
maior(6)
maior()

```

```
=====
Analisando os valores passados...
2 9 4 5 7 1 Foram informados 6 valores ao lado.
O maior valor informado foi 9.
=====
Analisando os valores passados...
4 7 0 Foram informados 3 valores ao lado.
O maior valor informado foi 7.
=====
Analisando os valores passados...
1 2 Foram informados 2 valores ao lado.
O maior valor informado foi 2.
=====
Analisando os valores passados...
6 Foram informados 1 valores ao lado.
O maior valor informado foi 6.
=====
Analisando os valores passados...
Foram informados 0 valores ao lado.
O maior valor informado foi 0.
```

## Lição 77 - Funções Para Sortear o Maior

In [54]:

```
from random import randint
from time import sleep

def sorteia(lista):
    print('Sorteando 5 valores da lista: ', end=' ')
    for cont in range(0, 5):
        n = randint(1, 10)
        lista.append(n)
        print(f'{n} ', end=' ', flush=True)
        sleep(0.3)
    print('PRONTO!')

def somaPar(lista):
    soma = 0
    for valor in lista:
        if valor % 2 == 0:
            soma += valor
    print(f'Somando os valores pares de {lista}, temos {soma}.')
    return soma

números = list()
sorteia(números)
somaPar(números)
```

Sorteando 5 valores da lista: 10 2 7 5 2 PRONTO!  
Somando os valores pares de [10, 2, 7, 5, 2], temos 14.

## Lição 78 - Funções (Parte 2)

In [55]:

```
def par(n=0):
    if n % 2 == 0:
        return True
    else:
```

```

        return False

num = int(input('Digite um número: '))
if par(num):
    print('É par!')
else:
    print('Não é par!')

```

É par!

## Lição 79 - Cores no Terminal

In [56]:

```

a = 3
b = 5
print('Os valores são \033[32m{}\033[m e \033[31m{}\033[m!!!'.format(a,

```

Os valores são 3 e 5!!!

Esse é um exemplo de uso de cores no terminal Python

## Lição 80 - Funções Para Votação

In [57]:

```

def voto(ano):
    from datetime import date
    atual = date.today().year
    idade = atual - ano
    if idade < 16:
        return f'Com {idade} anos: NÃO VOTA.'
    elif 16 <= idade < 18 or idade > 65:
        return f'Com {idade} anos: VOTO OPCIONAL.'
    else:
        return f'Com {idade} anos: VOTO OBRIGATÓRIO.'

# Programa principal
nasc = int(input("Em que ano você nasceu? "))
print(voto(nasc))

```

Com 20 anos: VOTO OBRIGATÓRIO.

## Lição 81 - Função Para Fatorial

In [58]:

```

def fatorial(n, show=False):
    """
    -> Calcula o Fatorial de um número.
    :param n: O número a ser calculado.
    :param show: (opcional) Mostrar ou não a conta.
    :return: O valor do Fatorial de um número n.
    """
    f = 1
    for c in range(n, 0, -1):
        if show:
            print(c, end=' ')
            if c > 1:
                print(' x ', end=' ')
        else:

```

```

        print(' = ', end=' ')
    f *= c
    return f

#Programa Principal
print(fatorial(5, show=True))
help(fatorial)

```

5 x 4 x 3 x 2 x 1 = 120  
Help on function fatorial in module \_\_main\_\_:

```

fatorial(n, show=False)
    -> Calcula o Fatorial de um número.
:param n: O número a ser calculado.
:param show: (opcional) Mostrar ou não a conta.
:return: O valor do Fatorial de um número n.

```

## Lição 82 - Ficha do Jogador

```
In [59]: def ficha(jog='desconhecido', gol=0):
    print(f'0 jogador {jog} fez {gol} gol(s) no campeonato.')

# Programa Principal
n = str(input("Nome do jogador: "))
g = str(input("Número de gols: "))
if g.isnumeric():
    g = int(g)
else:
    g = 0
if n.strip() == '':
    ficha(gol=g)
else:
    ficha(n, g)
```

0 jogador Paulo fez 0 gol(s) no campeonato.

## Lição 83 - Validando Entrada de Dados em Python

```
In [60]: def leiaInt(msg):
    ok = False
    valor = 0
    while True:
        n = str(input(msg))
        if n.isnumeric():
            valor = int(n)
            ok = True
        else:
            print('\033[0;31mERRO! Digite um número inteiro válido.\033')
        if ok:
            break
    return valor
```

```
# Programa Principal
n = leiaInt('Digite um número: ')
print(f'Você acabou de digitar o número {n}')
```

ERRO! Digite um número inteiro válido.  
Você acabou de digitar o número 2

## Lição 84 - Analisando e Gerando Dicionários

```
In [61]: def notas(*n, sit=False):
    """
    -> Função para analisar notas e situações de vários alunos.
    :param n: uma ou mais notas dos alunos (aceita várias).
    :param sit: valor opcional, indicado se deve ou não adicionar a situação.
    :return: dicionário com várias informações sobre a situação da turma.
    """
    r = dict()
    r['total'] = len(n)
    r['maior'] = max(n)
    r['menor'] = min(n)
    r['média'] = sum(n)/len(n)
    if sit:
        if r['média'] >= 7:
            r['situação'] = 'BOA'
        elif r['média'] >= 5:
            r['situação'] = 'RAZOÁVEL'
        else:
            r['situação'] = 'RUIM'
    return r

# Programa Principal
resp = notas(5.5, 2.5, 1.5, sit=True)
print(resp)
help(notas)

{'total': 3, 'maior': 5.5, 'menor': 1.5, 'média': 3.1666666666666665, 'situação': 'RUIM'}
Help on function notas in module __main__:

notas(*n, sit=False)
    -> Função para analisar notas e situações de vários alunos.
    :param n: uma ou mais notas dos alunos (aceita várias).
    :param sit: valor opcional, indicado se deve ou não adicionar a situação.
    :return: dicionário com várias informações sobre a situação da turma.
```

## Lição 85 - Interactive Helping System in Python

```
In [62]: from time import sleep
c = ('\033[m',           # 0 - sem cores
     '\033[0;30;41m',    # 1 - vermelho
     '\033[0;30;42m',    # 2 - verde
     '\033[0;30;43m',    # 3 - amarelo
     '\033[0;30;44m',    # 4 - azul
     '\033[0;30;45m',    # 5 - roxo
```

```

        '\033[7;30m'           # 6 - branco
    )

def ajuda(com):
    título(f'Acessando o manual do comando \'{com}\'', 4)
    print(c[6], end=' ')
    help(com)
    print(c[0], end=' ')
    sleep(2)

def título(msg, cor=0):
    tam = len(msg) + 4
    print(c[cor], end=' ')
    print('~' * tam)
    print(f' {msg} ')
    print('~' * tam)
    print(c[0], end=' ')
    sleep(1)

# Programa Principal
comando = ''
while True:
    título('SISTEMA DE AJUDA PyHELP', 2)
    comando = str(input("Função ou Biblioteca > "))
    if comando.upper() == 'FIM':
        break
    else:
        ajuda(comando)
    título('ATÉ LOGO!', 1)

```

SISTEMA DE AJUDA PyHELP

Acessando o manual do comando 'Função'

No Python documentation found for 'Função'.  
Use help() to get the interactive help utility.  
Use help(str) for help on the str class.

SISTEMA DE AJUDA PyHELP

Acessando o manual do comando 'Biblioteca'

No Python documentation found for 'Biblioteca'.  
Use help() to get the interactive help utility.  
Use help(str) for help on the str class.

SISTEMA DE AJUDA PyHELP

ATÉ LOGO!

# Lição X - Superconversor de Unidades

```
bb = n1 * 25 / 10000
bc = n1 * 25 / 100

bd = n1 * 1053 / 1000000
be = n1 * 4211 / 1000000
bf = n1 * 6316 / 1000000
bg = n1 * 1263 / 100000
bh = n1 * 2437 / 100000
bi = n1 * 2924 / 100000
bj = n1 * 1053 / 10000
bk = n1 * 2924 / 10000
bl = n1 * 4211 / 10000
bm = n1 * 1053 / 100

bn = n1 * 4249 / 10000000000
bo = n1 * 17 / 10000000
bp = n1 * 17 / 100000
bq = n1 * 3399 / 10000000
br = n1 * 6799 / 10000000
bs = n1 * 272 / 100000
bt = n1 * 2448 / 100000
bu = n1 * 979 / 10000
bv = n1 * 979 / 100
bw = n1 * 114 / 10

bx = n1 * 2925 / 10000000
by = n1 * 2925 / 100000
bz = n1 * 2632 / 10000
ca = n1 * 9477 / 1000
cb = n1 * 1365 / 1000

cc = n1 * 5725 / 100000000000
cd = n1 * 1431 / 1000000000
ce = n1 * 2588 / 1000000000
cf = n1 * 3106 / 1000000000
cg = n1 * 1553 / 10000000
ch = n1 * 2236 / 10000000
ci = n1 * 2544 / 10000000
cj = n1 * 1863 / 1000000
ck = n1 * 7454 / 1000000
cl = n1 * 8945 / 100000
cm = n1 * 1431 / 1000
cn = n1 * 1288 / 100

co = n1 * 2066 / 10000
cp = n1 * 8264 / 10000
cq = n1 * 2066 / 100
cr = n1 * 1322

cs = n1 * 9153 / 100000000
ct = n1 * 6865 / 100000000
cu = n1 * 8787 / 10000000000
cv = n1 * 1977 / 1000
cw = n1 * 1176 / 100

cx = n1 / 1000
cy = n1 / 1000
cz = n1 * 2446 / 1000000
da = n1 / 1000
db = n1 / 1000
```



```

print ('-' * 30)
print ('\033[0;30;43m-----GRÉCIA-----\033[m')
print ('-' * 30)
print ('\033[34m{} plethon {} aroura {} hektos {} hémiektos')

print ('-' * 30)
print ('\033[0;30;43m-----ROMA-----\033[m')
print ('-' * 30)
print ('\033[34m{} saltus {} centurium {} geredium {} uger {}')

print ('-' * 30)
print ('\033[0;30;43m-----FRANCÊS ANTIGO-----\033[m')
print ('-' * 30)
print ('\033[34m{} arpont {} perche {} toise quadrada {} pie {}')

print ('-' * 30)
print ('\033[0;30;43m-----ESPAÑHA ANTIGA-----\033[m')
print ('-' * 30)
print ('\033[34m{} legua {} labor {} caballería {} yugada {}')

print ('-' * 29)
print ('\033[0;30;43m-----PORTUGAL ANTIGO-----\033[m')
print ('-' * 29)
print ('\033[34m{} braça quadrada {} vara quadrada {} palmo {}')

print ('-' * 29)
print ('\033[0;30;43m-----RÚSSIA ANTIGA-----\033[m')
print ('-' * 29)
print ('\033[34m{} desyatina país {} desyatina fazendeiros {}')

print ('-' * 30)
print ('\033[0;30;46m-----AINDA EM USO-----\033[m')
print ('-' * 30)
print ('\033[36m{} dulum (дулум) da Bósnia, Herzegovina e Sérvia')

print ('-' * 29)
print ('\033[0;30;46m-----BITOLA DOS FIOS-----\033[m')
print ('-' * 29)
print ('\033[36m{} polegada circular {} kcmil, MCM {} cmil {}')

print ('-' * 30)
print ('\033[0;30;46m-----NATURAIS-----\033[m')
print ('-' * 30)
print ('\033[36m{} L2 |\033[m'.format(ec))

print ('\033[36mA CATEC agradece a sua consulta!\033[m')
case "2":
    n1 = float(input ('Digite um valor em \033[32ml\033[m: '))
    a = n1 / 100
    b = n1 / 10
    c = n1
    d = n1 * 10
    e = n1 * 100
    f = n1 * 1000
    g = n1 * 1000000
    h = n1 / 1000000000000000
    i = n1 / 1000
    j = n1
    k = n1 * 1000
    l = n1 * 1000000

```

```
m = n1 / 10000000000 * 8107
n = n1 / 100000 * 629
o = n1 / 10000 * 2642
p = n1 / 1000 * 1057
q = n1 / 1000 * 2113
r = n1 / 1000 * 8454
s = n1 / 100 * 3381
t = n1 / 10 * 2705
u = n1 * 16231

v = n1 / 1000000 * 8648
w = n1 / 100000 * 2838
x = n1 / 10000 * 1135
y = n1 / 1000 * 227
z = n1 / 10000 * 9081
aa = n1 / 1000 * 1816

ab = n1 / 1000 * 2113
ac = n1 / 1000 * 4227
ad = n1 / 1000 * 4227
ae = n1 / 1000 * 6763
af = n1 / 100 * 1691
ag = n1 / 100 * 3381
ah = n1 / 100 * 6763
ai = n1 / 10 * 1353
aj = n1 / 10 * 2705
ak = n1 / 10 * 2705
al = n1 * 16231

au = n1 / 1000000 * 1427
av = n1 / 100000 * 611
aw = n1 / 10000 * 275
ax = n1 / 100 * 11
ay = n1 / 100 * 22
az = n1 / 10000 * 8799
ba = n1 / 100 * 176
bb = n1 / 10 * 352
bc = n1 / 10 * 2816
bd = n1 / 10 * 8447
be = n1 * 16894

bf = n1 / 1000000000000000 * 2399
bg = n1 / 100000000 * 3531
bh = n1 / 1000000 * 1308
bi = n1 / 1000000 * 3531
bj = n1 / 100000 * 3531
bk = n1 / 100 * 6102

bl = n1 / 1000000 * 1047
bm = n1 / 1000000 * 2095
bn = n1 / 1000000 * 3142
bo = n1 / 100000 * 419
bp = n1 / 1000000 * 6285
bq = n1 / 100000 * 838
br = n1 / 100000 * 1466

bs = n1 / 1000000 * 1018
bt = n1 / 100000 * 611
bu = n1 / 100000 * 1222
```



```
ds = n1 * 5 / 10000
dt = n1 * 4 / 100
du = n1 * 5 / 100
dv = n1 * 1
dw = n1 * 8
dx = n1 * 32
dy = n1 * 128

dz = n1 / 100000 * 1957
ea = n1 / 10000 * 1174
eb = n1 / 10000 * 2348
ec = n1 / 10000 * 9393
ed = n1 / 1000 * 1879
ee = n1 / 1000 * 3757
ef = n1 / 100 * 1503
eg = n1 / 100 * 2254
eh = n1 / 10 * 2254

ei = n1 / 100000 * 2609
ej = n1 / 100000 * 3914
ek = n1 / 10000 * 3131
el = n1 / 1000 * 1879
em = n1 / 1000 * 3757
en = n1 / 1000 * 7514
eo = n1 / 100 * 1503
ep = n1 / 100 * 2254
eq = n1 / 100 * 4509
er = n1 / 100 * 9017
es = n1 / 10 * 1127
et = n1 / 10 * 2254

eu = n1 / 1000000 * 1928
ev = n1 / 100000 * 3856
ew = n1 / 100000 * 7712
ex = n1 / 100000 * 7712
ey = n1 / 10000 * 1157
ez = n1 / 10000 * 2314
fa = n1 / 10000 * 3085
fb = n1 / 1000 * 1851
fc = n1 / 1000 * 3702
fd = n1 / 1000 * 3702
fe = n1 / 1000 * 5552
ff = n1 / 1000 * 7403
fg = n1 / 10 * 111
fh = n1 / 100 * 1481
fi = n1 / 100 * 2221
fj = n1 / 100 * 8884

fk = n1 / 1000000 * 1502
fl = n1 / 100000 * 1802
fm = n1 / 100000 * 7209
fn = n1 / 10000 * 2163
fo = n1 / 10000 * 4325
fp = n1 / 10000 * 8651
fq = n1 / 10000 * 2163

fr = n1 / 1000000 * 3875
fs = n1 / 1000 * 62
ft = n1 / 1000 * 62
```

```
fu = n1 / 1000 * 496
fv = n1 / 1000 * 1323
fw = n1 / 1000 * 1984
fx = n1 / 1000 * 7937
fy = n1 / 1000 * 7943
fz = n1 / 1000 * 7937

ga = n1 / 1000000 * 1208
gb = n1 / 100000 * 1812
gc = n1 / 100000 * 7246
gd = n1 / 10000 * 2899
ge = n1 / 10000 * 5797
gf = n1 / 1000 * 1159
gg = n1 / 1000 * 2319
gh = n1 / 1000 * 4638
gi = n1 / 1000 * 9275

gj = n1 / 100000 * 119
gk = n1 / 1000000 * 2381
gl = n1 / 100000 * 5952
gm = n1 / 100000 * 5952
gn = n1 / 1000 * 119
go = n1 / 10000 * 7143
gp = n1 / 1000 * 2857
qq = n1 / 1000 * 5714
gr = n1 / 100 * 1143

gs = n1 / 100000 * 2941
gt = n1 / 1000 * 1203

gu = n1 / 100000 * 3811
gv = n1 / 10000 * 813
gw = n1 / 10000 * 3049

gx = n1 / 10000 * 813
gy = n1 / 1000 * 813
gz = n1 / 10000 * 3252
ha = n1 / 1000 * 1301
hb = n1 / 1000 * 1626
hc = n1 / 100 * 813
hd = n1 / 100 * 1626

he = n1 / 1000000 * 4545
hf = n1 / 1000000 * 9091
hg = n1 / 100000 * 4545
hh = n1 / 10000 * 1364
hi = n1 / 10000 * 8182
hj = n1 / 1000 * 3273
hk = n1 / 100 * 1964
hl = n1 / 100 * 3927

hm = n1 / 100000 * 4545
hn = n1 / 10000 * 2727
ho = n1 / 1000 * 3273

print ('-' * 30)
print ('\033[0;30;46m-----MEDIDAS MÉTRICAS-----\033[m')
print ('-' * 30)
print ('\033[36m{} hl | {} decalitro | {} l | {} dl | {} cl | {} ml
```

```
print ('-' * 30)
print ('\033[0;30;46m-----EUA LÍQUIDOS-----\033[m')
print ('-' * 30)
print ('\033[36m{} acre-pé {} barril (petróleo) {} gal {} qt')

print ('-' * 29)
print ('\033[0;30;46m-----EUA SECOS-----\033[m')
print ('-' * 29)
print ('\033[36m{} barril {} bu {} pk {} gal {} qt {} pt {}')

print ('-' * 30)
print ('\033[0;30;46m-----APOTHECARIES-----\033[m')
print ('-' * 30)
print ('\033[36m{} pt {} copo de vidro {} xícara de café da manhã')

print ('-' * 30)
print ('\033[0;30;46m-----INGLATERRA-----\033[m')
print ('-' * 30)
print ('\033[36m{} perch {} barril {} bu {} pk {} gal {} qt')

print ('-' * 30)
print ('\033[0;30;46m-----EUA/INGLATERRA-----\033[m')
print ('-' * 30)
print ('\033[36m{} mi³ {} mcf {} yd³ {} ccf {} ft³ {} in³')

print ('-' * 30)
print ('\033[0;30;46m-----INGLATERRA|VINHO-----\033[m')
print ('-' * 30)
print ('\033[36m{} tun {} butt {} puncheon {} hogsheads {}')

print ('-' * 30)
print ('\033[0;30;46m-----INGLATERRA|CERVEJA-----\033[m')
print ('-' * 30)
print ('\033[36m{} hogsheads {} barril {} kilderkin {} firkin {}')

print ('-' * 29)
print ('\033[0;30;46m-----EUA|CERVEJA-----\033[m')
print ('-' * 29)
print ('\033[36m{} Barril de cerveja norte-americano {} Meio-balde')

print ('-' * 29)
print ('\033[0;30;46m-----EUA~CULINÁRIA-----\033[m')
print ('-' * 29)
print ('\033[36m{} xícara {} colher de sopa {} colher de chá {}')

print ('-' * 29)
print ('\033[0;30;46m-----AUSTRÁLIA~CULINÁRIA-----\033[m')
print ('-' * 29)
print ('\033[36m{} colher de sopa {} colher de sobremesa {} colher de açúcar {}')

print ('-' * 29)
print ('\033[0;30;46m-----CULINÁRIA-----\033[m')
print ('-' * 29)
print ('\033[36m{} xícara {} colher de sopa {} colher de chá {}')

print ('-' * 29)
print ('\033[0;30;46m-----UNIDADES NATURAIS-----\033[m')
print ('-' * 29)
print ('\033[36m{} L³ |\033[m'.format(ck))
```

```
print ('-' * 30)
print ('\033[0;30;46m-----ASTRONOMIA-----\033[m')
print ('-' * 30)
print ('\033[36m{} pc³ |{} ano-luz cúbico |{} minuto-luz cúbico |')

print ('-' * 29)
print ('\033[0;30;46m-----NAVAL-----\033[m')
print ('-' * 29)
print ('\033[36m{} arqueação |{} tonelada de registro |{} Sister |')

print ('-' * 29)
print ('\033[0;30;46m-----MADEIRA-----\033[m')
print ('-' * 29)
print ('\033[36m{} milhão de board-feet (mmfbm, mmbdft, mmbf) |')

print ('-' * 29)
print ('\033[0;30;46m-----JAPÃO-----\033[m')
print ('-' * 29)
print ('\033[36m{} sai |{} shaku |{} go |{} sho |{} to |{} koku |')

print ('-' * 29)
print ('\033[0;30;46m-----CHINA 1915#1930-----\033[m')
print ('-' * 29)
print ('\033[36m{} dan |{} hu |{} dou |{} sheng |{} ge |{} shao |')

print ('-' * 29)
print ('\033[0;30;46m-----CHINA 1930#HOJE-----\033[m')
print ('-' * 29)
print ('\033[36m{} dan |{} dou |{} sheng |{} ge |{} shao |{} cu |')

print ('-' * 29)
print ('\033[0;30;46m-----TAILÂNDIA-----\033[m')
print ('-' * 29)
print ('\033[36m{} kwian |{} sat |{} thang |{} thanan |{} fai mi |')

print ('-' * 29)
print ('\033[0;30;43m-----GRÉCIA ÁTICO SECO-----\033[m')
print ('-' * 29)
print ('\033[34m{} medimnos |{} hekteus |{} hēmiekton |{} choineion |')

print ('-' * 30)
print ('\033[0;30;43m-----GRÉCIA ÁTICO LÍQUIDO-----\033[m')
print ('-' * 30)
print ('\033[34m{} metrētēs |{} keramion |{} chous |{} xestēs |')

print ('-' * 30)
print ('\033[0;30;43m-----ROMA-----\033[m')
print ('-' * 30)
print ('\033[34m{} culeus |{} amphora quadrantal |{} urna |{} medimnos |')

print ('-' * 29)
print ('\033[0;30;43m-----ESPAÑHA ANTIGA SECO-----\033[m')
print ('-' * 29)
print ('\033[34m{} cahíz |{} fanega |{} cuartilla |{} celemín |')

print ('-' * 30)
print ('\033[0;30;43m-----ESPAÑHA ANTIGA LÍQUIDO-----\033[m')
print ('-' * 30)
print ('\033[34m{} moyo |{} cántara |{} arroba |{} azumbre |{} medidors |')
```

```

print ('-' * 30)
print ('\033[0;30;43m----PORTUGAL ANTIGO SECO----\033[m')
print ('-' * 30)
print ('\033[34m{} moio {} fanga {} alqueire {} quarta {} o:')

print ('-' * 29)
print ('\033[0;30;43m--PORTUGAL ANTIGO LÍQUIDO---\033[m')
print ('-' * 29)
print ('\033[34m{} tonel {} pipa {} almude {} cântaro {} po:')

print ('-' * 30)
print ('\033[0;30;43m-----FRANCÊS ANTIGO-----\033[m')
print ('-' * 30)
print ('\033[34m{} minot {} litron |\033[m'.format(gs, gt))

print ('-' * 30)
print ('\033[0;30;43m-----RÚSSIA ANTIGA SECA-----\033[m')
print ('-' * 30)
print ('\033[34m{} cetverik {} vedro {} garnetz |\033[m'.format(gs, gt))

print ('-' * 29)
print ('\033[0;30;43m---RÚSSIA ANTIGA LÍQUIDA---\033[m')
print ('-' * 29)
print ('\033[34m{} vedro {} shtoff {} chetvert {} garrafa de:')

print ('-' * 29)
print ('\033[0;30;42m-----BÍBLIA SECO-----\033[m')
print ('-' * 29)
print ('\033[35m{} coro {} letec {} efa {} seá {} cabe {} :')

print ('-' * 30)
print ('\033[0;30;42m-----BÍBLIA LÍQUIDO-----\033[m')
print ('-' * 30)
print ('\033[35m{} bato {} him {} log |\033[m'.format(hm, hn, lg))

print ('\033[36mA CATEC agradece a sua consulta!\033[m')
case "3":
    n1 = float(input ('Digite um valor em \033[32mkm\033[m: '))
    a = n1
    b = n1 * 10
    c = n1 * 100
    d = n1 * 1000
    e = n1 * 10000
    f = n1 * 100000
    g = n1 * 1000000
    h = n1 * 1000000000
    i = n1 * 1000000000000
    j = n1 * 1000000000000000

    k = n1 / 10000 * 2071
    l = n1 / 10000 * 6214
    m = n1 / 10000 * 6214
    n = n1 / 1000 * 4971
    o = n1 / 100 * 2734
    p = n1 / 100 * 4971
    q = n1 / 10 * 1988
    r = n1 / 10 * 1988
    s = n1 / 10 * 1988
    t = n1 * 1094
    u = n1 * 3281

```

```
v = n1 * 4374
w = n1 * 9843
x = n1 * 39370
y = n1 * 472441
z = n1 * 39370079
aa = n1 * 39370078740

ab = n1 / 10000 * 2071
ac = n1 / 10000 * 6214
ad = n1 / 10000 * 6214
ae = n1 / 1000 * 9113
af = n1 / 100 * 2734
ag = n1 / 100 * 3645
ah = n1 / 1000 * 4971
ai = n1 / 100 * 4971
aj = n1 * 164
ak = n1 / 10 * 1988
al = n1 / 10 * 7291
au = n1 / 10 * 8749
av = n1 * 1094
aw = n1 * 1312
ax = n1 * 2187
ay = n1 * 3281
az = n1 * 4374
ba = n1 * 4374
bb = n1 * 6562
bc = n1 * 9843
bd = n1 * 13123
be = n1 * 39370
bf = n1 * 44994
bg = n1 * 52494
bh = n1 * 118110
bi = n1 * 472441
bj = n1 * 472441
bk = n1 * 472441
bl = n1 * 3937015
bm = n1 * 39370147
bn = n1 * 39370147

bo = n1 / 1000 * 4971
bp = n1 / 100 * 3281
bq = n1 / 100 * 4971
br = n1 / 100 * 9942
bs = n1 / 10 * 1988
bt = n1 * 1094
bu = n1 * 3281
bv = n1 * 4971
bw = n1 * 4971
bx = n1 * 3281

by = n1 / 100 * 18
bz = n1 / 100 * 54
ca = n1 / 10 * 54
cb = n1 / 10 * 5468

cc = n1 / 10000 * 5396
cd = n1 / 1000 * 4557
ce = n1 / 10 * 5468

cf = n1 / 10000 * 5396
```

```
cg = n1 / 1000 * 5396
ch = n1 / 1000 * 7698
ci = n1 * 10
cj = n1 / 1000 * 7975
ck = n1 / 10 * 125
cl = n1 / 10 * 119
cm = n1 / 10 * 119
cn = n1 / 100 * 1154
co = n1 / 100 * 1164
cp = n1 * 1155
cq = n1 / 10 * 9091
cr = n1 * 1196
cs = n1 * 1250
ct = n1 * 1196
cu = n1 * 1179
cv = n1 * 1196
cw = n1 * 1190
cx = n1 * 1193
cy = n1 * 1190
cz = n1 * 1198
da = n1 * 1193
db = n1 * 1190
dc = n1 * 1250
dd = n1 * 1151
de = n1 * 1193
df = n1 * 1196
dg = n1 * 1164
dh = n1 * 1190
di = n1 * 1196

dj = n1 / 10000 * 2546
dk = n1 / 1000 * 9167
dl = n1 * 330
dm = n1 * 550
dn = n1 * 550
do = n1 * 3300
dp = n1 * 33000
dq = n1 * 330000
dr = n1 * 3300000
ds = n1 * 33000000

dt = n1 / 10 * 1736
du = n1 / 100 * 3125
dv = n1 / 10 * 3125
dw = n1 * 625
dx = n1 * 3125
dy = n1 * 31250
dz = n1 * 312500
ea = n1 * 3125000
eb = n1 * 31250000
ec = n1 * 312500000
ed = n1 * 3125000000

ee = n1 * 2
ef = n1 * 30
eg = n1 * 300
eh = n1 * 3000
ei = n1 * 30000
ej = n1 * 300000
```



```
gg = n1 * 10667
gh = n1 * 13333
gi = n1 * 53333

gj = n1 / 10000 * 1349
gk = n1 / 10000 * 1799
gl = n1 / 10000 * 4498
gm = n1 / 10000 * 6747
gn = n1 / 1000 * 1349
go = n1 / 1000 * 2699
gp = n1 / 1000 * 5397
gq = n1 / 1000 * 5201
gr = n1 / 1000 * 5406
gs = n1 / 1000 * 5405
gt = n1 / 100 * 3238
gu = n1 / 100 * 5397
gv = n1 / 10 * 3238
gw = n1 / 10 * 3238
gx = n1 / 10 * 3238
gy = n1 / 10 * 5397
gz = n1 / 10 * 6477
ha = n1 * 1295
hb = n1 * 2159
hc = n1 * 2591
hd = n1 * 2879
he = n1 * 3238
hf = n1 * 4318
hg = n1 * 4710
hh = n1 * 5181
hi = n1 * 6477
hj = n1 * 6477
hk = n1 * 12953
hl = n1 * 12953
hm = n1 * 25907
hn = n1 * 51813

ho = n1 / 10000 * 1689
hp = n1 / 10000 * 4505
hq = n1 / 10000 * 6757
hr = n1 / 1000 * 5405
hs = n1 / 100 * 2815
ht = n1 / 10 * 3378
hu = n1 / 10 * 3378
hv = n1 / 10 * 6757
hw = n1 * 1351
hx = n1 * 1351
hy = n1 * 2252
hz = n1 * 2703
ia = n1 * 3378
ib = n1 * 4505
ic = n1 * 13514
id = n1 * 40541
ie = n1 * 40541
ig = n1 * 54054

ih = n1 / 10000 * 1495
ii = n1 / 10000 * 1794
ij = n1 / 10000 * 2393
ik = n1 / 10000 * 5383
il = n1 / 10000 * 7178
```

```
im = n1 / 100 * 2991
io = n1 * 145
ip = n1 / 10 * 2991
iq = n1 / 10 * 2991 * 2
ir = n1 / 10 * 7178
it = n1 * 1196
iu = n1 * 1794
iv = n1 * 2393
iw = n1 * 3589
ix = n1 * 4785
iy = n1 * 9570
iz = n1 * 43067
ja = n1 * 516805
jb = n1 * 6201662

jc = n1 / 10 * 4545
jd = n1 / 10 * 7576
je = n1 / 10 * 6061
jf = n1 / 10 * 9091
jg = n1 * 1515
jh = n1 * 1515 * 2
ji = n1 * 1515 * 3
jj = n1 * 36364
jk = n1 * 436364
jl = n1 * 5236364

jm = n1 / 10000 * 2246
jn = n1 / 10000 * 2252
jo = n1 / 10000 * 2565
jp = n1 / 100 * 25
jq = n1 / 10 * 171
jr = n1 * 171
js = n1 / 10 * 5131
jt = n1 * 3078
ju = n1 * 36941
jv = n1 * 443295

jw = n1 / 10000 * 1339
jx = n1 / 10000 * 9374
jy = n1 / 10000 * 4687
jz = n1 / 10 * 4032
ka = n1 / 10 * 4687
kb = n1 / 10 * 5682
kc = n1 * 1406
kd = n1 * 2381
ke = n1 * 3281
kf = n1 * 5624
kg = n1 * 22497
kh = n1 * 39370
ki = n1 * 393701

kj = n1 / 10000 * 1328
kk = n1 / 10 * 2655
kl = n1 * 1499
km = n1 * 3186
kn = n1 * 38233
ko = n1 * 458793

kp = n1 / 10000 * 1318
kq = n1 / 10 * 2636
```

```
kr = n1 / 10 * 5273
ks = n1 * 3164
kt = n1 * 37965
ku = n1 * 455581
kv = n1 * 5466970

kw = n1 / 10
kx = n1 / 100000 * 9356
ky = n1 / 10000 * 3742
kz = n1 / 100 * 2105
la = n1 / 100 * 3368
lb = n1 / 10 * 2105
lc = n1 / 10 * 5614
ld = n1 * 1684
le = n1 * 3368
lf = n1 * 6736
lg = n1 * 10104
lh = n1 * 40418
li = n1 * 33681
lj = n1 * 485012
lk = n1 * 404176

ll = n1 * 2187
lm = n1 * 4374
ln = n1 * 13123
lo = n1 * 52493
print ('-' * 30)
print ('\033[0;30;46m-----MEDIDAS MÉTRICAS-----\033[m')
print ('-' * 30)
print ('\033[36m{} km |{} hm |{} dam |{} m |{} dm |{} cm |{} mm')

print ('-' * 30)
print ('\033[0;30;46m-----EUA E INGLATERRA-----\033[m')
print ('-' * 30)
print ('\033[36m{} léguia |{} mi |{} land |{} furlong |{} bolt |')

print ('-' * 30)
print ('\033[0;30;46m-----INGLATERRA-----\033[m')
print ('-' * 30)
print ('\033[36m{} léguia |{} mi |{} land |{} skein |{} bolt |')

print ('-' * 29)
print ('\033[0;30;46m-----AGRIMENSURA DOS EUA-----\033[m')
print ('-' * 29)
print ('\033[36m{} furlong |{} corrente de ramsden - engenharia')

print ('-' * 29)
print ('\033[0;30;46m-----NÁUTICA-----\033[m')
print ('-' * 29)
print ('\033[36m{} naut.leg |{} naut.mi |{} cbl |{} fath |')

print ('-' * 29)
print ('\033[0;30;46m-----NÁUTICA EUA-----\033[m')
print ('-' * 29)
print ('\033[36m{} milha marítima norte-americana |{} cabo norte')

print ('-' * 30)
print ('\033[0;30;46m-----NÁUTICA INGLATERRA-----\033[m')
print ('-' * 30)
print ('\033[36m{} mira do almirantado |{} cabo do almirantado')
```

```
print ('-' * 29)
print ('\033[0;30;46m-----ESPAÑA-----\033[m')
print ('-' * 29)
print ('\033[36m{} cuadra da Argentina |{} cuadra da Bolívia |{}')

print ('-' * 29)
print ('\033[0;30;46m-----JAPÃO-----\033[m')
print ('-' * 29)
print ('\033[36m{} ri |{} cho |{} jo |{} hiro |{} ken |{} shaku')

print ('-' * 29)
print ('\033[0;30;46m-----CHINA 1915#1930-----\033[m')
print ('-' * 29)
print ('\033[36m{} li - 里 |{} yin |{} zhang |{} bu |{} chi |{}')

print ('-' * 29)
print ('\033[0;30;46m-----CHINA 1930#HOJE-----\033[m')
print ('-' * 29)
print ('\033[36m{} li - 里 |{} yin |{} zhang |{} chi |{} cun |{}')

print ('-' * 29)
print ('\033[0;30;46m-----HONG KONG-----\033[m')
print ('-' * 29)
print ('\033[36m{} cek |{} tsun |{} fan |{}\033[m'.format(ep, eq))

print ('-' * 29)
print ('\033[0;30;46m-----TAILÂNDIA-----\033[m')
print ('-' * 29)
print ('\033[36m{} yot |{} sen |{} wa |{} sok |{} khuep |{} nio')

print ('-' * 30)
print ('\033[0;30;46m-----ÓPTICA-----\033[m')
print ('-' * 30)
print ('\033[36m{} dioptria |{}\033[m'.format(ez))

print ('-' * 29)
print ('\033[0;30;46m-----GEOGRAFIA ALEMÃ-----\033[m')
print ('-' * 29)
print ('\033[36m{} milha geográfica |{}\033[m'.format(fa))

print ('-' * 30)
print ('\033[0;30;46m-----COMPUTAÇÃO-----\033[m')
print ('-' * 30)
print ('\033[36m{} U |{} HP |{}\033[m'.format(fb, fc))

print ('-' * 29)
print ('\033[0;30;46m-----ATA-----\033[m')
print ('-' * 29)
print ('\033[36m{} paica |{} paica - PostScript |{} ponto |{} po')

print ('-' * 29)
print ('\033[0;30;46m-----DIDOT-----\033[m')
print ('-' * 29)
print ('\033[36m{} cícero |{} ponto |{}\033[m'.format(fj, fk))

print ('-' * 29)
print ('\033[0;30;46m-----CALÇADO EUROPEU-----\033[m')
print ('-' * 29)
print ('\033[36m{} ponto parisiense |{}\033[m'.format(fl))
```

```
print ('-' * 30)
print ('\033[0;30;46m-----ASTRONOMIA-----\033[m')
print ('-' * 30)
print ('\033[36m{} z {} pc {} ano-luz {} au {} minuto-luz {}')

print ('-' * 30)
print ('\033[0;30;46m-----GRACE HOPPER-----\033[m')
print ('-' * 30)
print ('\033[36m{} microsegundo-luz {} nanosegundo-luz {} p:')

print ('-' * 30)
print ('\033[0;30;46m-----NATURAIS-----\033[m')
print ('-' * 30)
print ('\033[36m{} L |\033[m'.format(fv))

print ('-' * 29)
print ('\033[0;30;43m-----EGITO-----\033[m')
print ('-' * 29)
print ('\033[34m{} iteru {} knet {} nbiw {} meh niswt {} mel {}')

print ('-' * 30)
print ('\033[0;30;43m-----GRÉCIA-----\033[m')
print ('-' * 30)
print ('\033[34m{} schoinos {} parasanges {} dolichos {} míl: {}')

print ('-' * 30)
print ('\033[0;30;43m-----ROMA-----\033[m')
print ('-' * 30)
print ('\033[34m{} schœnus {} leuga {} mille passus {} stadii {}')

print ('-' * 30)
print ('\033[0;30;43m-----ESPAÑHA ANTIGA-----\033[m')
print ('-' * 30)
print ('\033[34m{} legua de por grado {} legua marina {} legua {}')

print ('-' * 29)
print ('\033[0;30;43m-----PORTUGAL ANTIGO-----\033[m')
print ('-' * 29)
print ('\033[34m{} braça {} toesa {} passo geométrico {} varas {}')

print ('-' * 30)
print ('\033[0;30;43m-----FRANCÊS ANTIGO-----\033[m')
print ('-' * 30)
print ('\033[34m{} lieue commune {} lieue marine {} lieue de {}')

print ('-' * 29)
print ('\033[0;30;43m-----RÚSSIA ANTIGA-----\033[m')
print ('-' * 29)
print ('\033[34m{} milha {} verst {} mezhevaya verst {} kosa {}')

print ('-' * 29)
print ('\033[0;30;43m-----PRÚSSIA-----\033[m')
print ('-' * 29)
print ('\033[34m{} meile {} ruthe {} elle {} fuss {} zoll {}')

print ('-' * 30)
print ('\033[0;30;43m-----ÁUSTRIA ANTIGA-----\033[m')
print ('-' * 30)
print ('\033[34m{} meile {} ruthe {} klafner {} fuss {} zol: {}')
```

```
print ('-' * 29)
print ('\033[0;30;43m-----SUÉCIA ANTIGA-----\033[m')
print ('-' * 29)
print ('\033[34m{} nymil |{} milha |{} fjärdingsväg |{} rev ante')

print ('-' * 30)
print ('\033[4;30;42m-----BÍBLIA-----\033[m')
print ('-' * 30)
print ('\033[35m{} côvado |{} palmo |{} tephach |{} etzbah |\033[m')

print ('\033[36mA CATEC agradece a sua consulta!\033[m')
case "4":
    n1 = float(input ('Digite um valor em \033[32mkg\033[m: '))
    a = n1 / 1000000
    b = n1 / 1000
    c = n1
    d = n1 * 1000
    e = n1 * 100000
    f = n1 * 1000000
    g = n1 * 1000000000
    h = n1 / 1000 * 9807
    i = n1 / 1000000 * 9807
    j = n1 * 5000
    k = n1 / 100
    l = n1 / 100
    m = n1 * 602214129011674000000000000000
    n = n1 / 100000 * 6808
    o = n1 / 100000 * 1261
    p = n1 / 100000 * 6808 / 4
    q = n1 / 100000 * 6808 * 32
    r = n1 / 100000 * 6808 * 128
    s = n1 / 100000 * 6808 * 512
    t = n1 / 100000 * 6808 * 4096
    u = n1 / 100000 * 6808 * 12288
    v = n1 / 100000 * 6808 * 294912
    w = n1 / 1000000 * 3623
    x = n1 / 100000 * 3212
    y = n1 / 100000 * 2415
    z = n1 / 100000 * 3105
    aa = n1 / 100000 * 3623
    ab = n1 / 10000 * 1449
    ac = n1 / 1000 * 2174
    ad = n1 / 1000 * 2174 * 2
    ae = n1 / 1000 * 2174 * 4
    af = n1 / 1000 * 2174 * 16
    ag = n1 / 1000 * 2174 * 128
    ah = n1 / 1000 * 2174 * 256
    ai = n1 / 1000 * 2174 * 9216
    aj = n1 / 100000 * 2043
    ak = n1 / 1000 * 2043
    al = n1 / 1000000 * 6105
    au = n1 / 1000000 * 6105 * 10
    av = n1 / 1000000 * 6105 * 400
    aw = n1 / 1000000 * 6105 * 12800
    ax = n1 / 1000000 * 6105 * 38400
    ay = n1 / 1000000 * 6105 * 3686400
    az = n1 / 1000000 * 6479
    ba = n1 / 1000000 * 9719
    bb = n1 / 100000 * 1944
```

```
bc = n1 / 100000 * 9719
bd = n1 / 1000 * 2138
be = n1 / 1000 * 2138 * 2
bf = n1 / 100 * 3421
bg = n1 / 100 * 3421 * 2
bh = n1 / 10 * 2053
bi = n1 / 10 * 2737
bj = n1 * 1095
bk = n1 * 16420
bl = n1 / 1000000 * 4464
bm = n1 / 1000000 * 6493
bn = n1 / 100000 * 1786
bo = n1 / 100000 * 8117
bp = n1 / 1000 * 1786
bq = n1 / 1000 * 1786 * 2
br = n1 / 1000 * 1786 * 4
bs = n1 / 1000 * 1786 * 16
bt = n1 / 100 * 6842
bu = n1 / 100 * 6842 * 4
bv = n1 / 100 * 6842 * 16
bw = n1 / 1000000 * 5881
bx = n1 / 1000000 * 5881 * 20
by = n1 / 1000 * 196
bz = n1 / 1000000 * 5881 * 400
ca = n1 / 1000000 * 5881 * 800
cb = n1 / 1000000 * 5881 * 40000
cc = n1 / 1000000 * 1102
cd = n1 / 1000000 * 1102 * 2000
ce = n1 / 1000000 * 1102 * 32000
cf = n1 / 10 * 2746
cg = n1 * 5217
ch = n1 / 100000 * 2939
ci = n1 / 100000 * 2939 * 60
cj = n1 / 100000 * 2939 * 3000
ck = n1 / 100000 * 2939 * 4500
cl = n1 / 100000 * 2939 * 6000
cm = n1 / 100000 * 2939 * 60000
cn = n1 / 100 * 3649 / 12
co = n1 / 100 * 3649
cp = n1 / 100 * 3649 * 2
cq = n1 / 100 * 3649 * 3
cr = n1 / 100 * 3649 * 4
cs = n1 / 100 * 3649 * 6
ct = n1 / 100 * 3649 * 8
cu = n1 / 100 * 3649 * 12
cv = n1 / 100 * 3649 * 24
cw = n1 / 100 * 3649 * 48
cx = n1 / 100 * 3649 * 144
cy = n1 / 1000 * 3317
cz = n1 / 1000 * 3649
da = n1 / 1000 * 4054
db = n1 / 1000 * 4561
dc = n1 / 1000 * 5212
dd = n1 / 100 * 3649 / 6
de = n1 / 1000 * 7297
df = n1 / 100 * 3649 / 4
dg = n1 / 100 * 3649 / 3
dh = n1 / 100 * 3649 / 3
di = n1 / 100 * 3649 / 2
dj = n1 / 100 * 2432
```

```
dk = n1 / 100000 * 2646
dl = n1 / 100000 * 2646 * 60
dm = n1 / 100000 * 2646 * 600
dn = n1 / 100000 * 2646 * 1500
do = n1 / 100000 * 2646 * 3000
dp = n1 / 100000 * 2646 * 3000
dq = n1 / 100000 * 2646 * 6000
dr = n1 / 100000 * 2646 * 9000
ds = n1 / 100000 * 2646 * 12000
dt = n1 / 100000 * 2646 * 18000
du = n1 / 100000 * 2646 * 36000
dv = n1 / 100000 * 2646 * 48000
dw = n1 / 100000 * 2646 * 72000
dx = n1 / 100000 * 2646 * 24000
dy = n1 / 100000 * 2646 * 144000
dz = n1 / 100000 * 2646 * 288000
ea = n1 / 100000 * 3858
eb = n1 / 100000 * 3858 * 60
ec = n1 / 100000 * 3858 * 600
ed = n1 / 100000 * 3858 * 1500
ee = n1 / 100000 * 3858 * 3000
ef = n1 / 100000 * 3858 * 3000
eg = n1 / 100000 * 3858 * 6000
eh = n1 / 100000 * 3858 * 9000
ei = n1 / 100000 * 3858 * 12000
ej = n1 / 100000 * 3858 * 18000
ek = n1 / 100000 * 3858 * 36000
el = n1 / 100000 * 3858 * 48000
em = n1 / 100000 * 3858 * 72000
en = n1 / 100000 * 3858 * 24000
eo = n1 / 100000 * 3858 * 144000
ep = n1 / 100000 * 3858 * 288000
eq = n1 * 45945087
er = n1 / 100000 * 1667
es = n1 / 100000 * 1667 * 50
et = n1 / 100000 * 1667 * 1000
eu = n1 / 100000 * 1667 * 4000
ev = n1 / 100 * 6596
ew = n1 / 100000 * 1667 * 8000
ex = n1 / 100000 * 1667 * 16000
ey = n1 / 100000 * 1667 * 32000
ez = n1 / 100000 * 1667 * 64000
fa = n1 / 100000 * 1667 * 128000
fb = n1 / 100000 * 1667 * 256000
fc = n1 / 100000 * 1667 * 512000
fd = n1 / 100000 * 1667 * 25600000
fe = n1 / 100000 * 134
ff = n1 / 100000 * 134 * 20
fg = n1 / 100000 * 134 * 160
fh = n1 / 100000 * 134 * 640
fi = n1 / 100000 * 134 * 800
fj = n1 / 100000 * 134 * 3200
fk = n1 / 100000 * 134 * 12800
fl = n1 / 100000 * 134 * 2400
fm = n1 / 100000 * 134 * 57600
fn = n1 / 100000 * 134 * 64000
fo = n1 / 100000 * 134 * 768000
fp = n1 / 100000 * 134 * 6144000
fq = n1 / 10000 * 2667
fr = n1 / 1000 * 1667
```

```
fs = n1 / 10000 * 2667 * 10
ft = n1 / 10000 * 2667 * 1000
fu = n1 / 10000 * 2667 * 10000
fv = n1 / 100000 * 1653
fw = n1 / 1000 * 1653
fx = n1 / 100 * 2646
fy = n1 / 10 * 2646
fz = n1 * 2646
ga = n1 / 100 * 2672
gb = n1 / 10 * 2672
gc = n1 * 2672
gd = n1 / 1000 * 1676
ge = n1 / 1000 * 1676 * 16
gf = n1 / 1000 * 1676 * 160
gg = n1 / 1000 * 1676 * 1600
gh = n1 / 1000 * 1676 * 16000
gi = n1 / 1000 * 1676 * 160000
gj = n1 / 100 * 2
gk = n1 * 2
gl = n1 * 32
gm = n1 * 320
gn = n1 * 3200
go = n1 * 32000
gp = n1 * 320000
gq = n1 * 3200000
gr = n1 / 100 * 2
gs = n1 * 2
gt = n1 * 20
gu = n1 * 200
gv = n1 * 2000
gw = n1 * 20000
gx = n1 * 200000
gy = n1 * 2000000
gz = n1 * 20000000
ha = n1 / 10000000 * 9842
hb = n1 / 1000000 * 1102
hc = n1 / 1000000 * 2205
hd = n1 / 100000 * 1968
he = n1 / 100000 * 1968
hf = n1 / 100000 * 2205
hg = n1 / 10000 * 1575
hh = n1 / 1000 * 2205
hi = n1 / 100 * 3527
hj = n1 / 10 * 5644
hk = n1 * 15432
hl = n1 / 1000 * 2679
hm = n1 / 100 * 3215
hn = n1 * 643
ho = n1 * 4878
hp = n1 * 15432
hq = n1 * 308647
hr = n1 * 7407532
hs = n1 / 1000 * 2679
ht = n1 / 100 * 3215
hu = n1 / 10 * 2572
hv = n1 / 10 * 7716
hw = n1 * 15432
print ('{} kT {} T {} kg {} g {} cg {} mg {} mcg'.format(a, b, c, d, e, f, g, h, i))
print ('{} N {} kN'.format(h, i))
print ('{} ct'.format(j))
```

```

print ('{} centner métrico {} quintal métrico'.format(k, l))
print ('{} amu'.format(m))
print ('{} M'.format(eq))
print ('-----APOTHECARIES-----')
print ('{} lb {} onça {} dracma {} escrópulo {} grão'.format(hs))
print ('-----TROY-----')
print ('{} lb {} ozt {} dwt {} ct {} grão {} mite {} doite'.format(cc))
print ('-----AVOIRDUPOIS-----')
print ('{} T longa INGLATERRA {} T curta EUA {} klb Massa {} qu'.format(cc))
print ('-' * 30)
print ('\033[0;30;43m-----PORTUGAL-----\033[m')
print ('-' * 30)
print ('\033[34m{} @ |{} T |{} quintal |{} arrátel |{} quarta |'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----ESPAÑHA-----\033[m')
print ('-' * 29)
print ('\033[34m{} T |{} alqueire de trigo |{} alqueire de cente'.format(cc))
print ('-' * 30)
print ('\033[0;30;43m-----FRANÇA-----\033[m')
print ('-' * 30)
print ('\033[34m{} quintal |{} £ |{}\033[m'.format(aj, ak))
print ('-' * 30)
print ('\033[0;30;43m-----RÚSSIA-----\033[m')
print ('-' * 30)
print ('\033[34m{} berkovets |{} pud |{} £ |{} lot |{} zolotnik'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----PRÚSSIA-----\033[m')
print ('-' * 29)
print ('\033[34m{} schiffspfund |{} doppelzentner |{} zentner |'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----ÁUSTRIA-----\033[m')
print ('-' * 29)
print ('\033[34m{} kauch |{} saum |{} zentner |{} stain |{} pfui'.format(cc))
print ('-' * 30)
print ('\033[0;30;43m-----SUÉCIA-----\033[m')
print ('-' * 30)
print ('\033[34m{} skeppspund |{} lispund |{} bismerpund |{} sk'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----CHINA 1915#1930-----\033[m')
print ('-' * 29)
print ('\033[34m{} jin |{} liang |{} qian |{} fen |{} li |{} ha'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----CHINA 1930#1959-----\033[m')
print ('-' * 29)
print ('\033[34m{} dan |{} jin |{} liang |{} qian |{} fen |{} l'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----CHINA 1959#HOJE-----\033[m')
print ('-' * 29)
print ('\033[34m{} dan |{} jin |{} liang |{} qian |{} fen |{} l'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----EUA-----\033[m')
print ('-' * 29)
print ('\033[34m{} T curta |{} lb |{} oz |{}\033[m'.format(cc, cd))
print ('-' * 29)
print ('\033[0;30;43m-----HONG KONG-----\033[m')
print ('-' * 29)
print ('\033[34m{} picul |{} catty |{} tael |{} mace |{} candare'.format(cc))
print ('-' * 29)
print ('\033[0;30;43m-----JAPÃO-----\033[m')
print ('-' * 29)

```



```

n = n1 / 10 * 6562
o = n1 / 10000 * 6214
p = n1 * 3281
q = n1 * 39370
r = n1 * 5447
s = n1 * 28759228
t = n1 * 345110740
u = n1 * 60 / e / e
v = n1 * 3600 / e / e
w = n1 * 360 / e / e
x = n1 / 100 * 9656 / e / e
y = n1 * 5794 / e / e
z = n1 / 10 * 3292 / e / e
aa = n1 * 1800 / e / e
ab = n1 / 100 * 54
ac = n1 / 100 * 54
ad = n1 / 1000000 * 817
ae = n1 / 1000000000000 * 926
print ('-' * 30)
print ('\033[0;30;43m-----MEDIDAS MÉTRICAS-----\033[m')
print ('-' * 30)
print ('\033[34m{} km/s |{} m/s |{} quilômetro por minuto |{} milha por hora-----\033[m')
print ('-' * 30)
print ('\033[0;30;43m-----EUA E INGLATERRA-----\033[m')
print ('-' * 30)
print ('\033[34m{} milha por segundo |{} pé por segundo |{} polegada por segundo-----\033[m')
print ('-' * 29)
print ('\033[0;30;43m-----RITMO-----\033[m')
print ('-' * 29)
print ('\033[34m{} minutos por quilômetro |{} segundos por quilômetro-----\033[m')
print ('-' * 29)
print ('\033[0;30;43m-----UNIDADES NÁUTICAS-----\033[m')
print ('-' * 29)
print ('\033[34m{} nó |{} milha marítima por hora |\033[m.formato de saída')
print ('-' * 30)
print ('\033[4;30;42m-----FÍSICA-----\033[m')
print ('-' * 30)
print ('\033[35m{} velocidade do som no ar |{} velocidade da luz-----\033[m')

print ('\033[36mA CATEC agradece a sua consulta!\033[m')
case _:
    print('\033[31mTivemos um problema com os tipos de dados que voce inseriu')

```

Escolhas:

- 1 - Área
- 2 - Capacidade
- 3 - Comprimento
- 4 - Peso
- 5 - Temperatura
- 6 - Velocidade

----- MEDIDAS MÉTRICAS -----

5.0 m<sup>2</sup> | 500.0 dm<sup>2</sup> | 50000.0 cm<sup>2</sup> | 5000000.0 mm<sup>2</sup> | 5e+28 b | 5e-06 km<sup>2</sup> | 0.000  
5 hectare | 0.005 decare | 0.05 a |

----- EUA E INGLATERRA -----

5.365e-08 township | 1.9305e-06 milha quadrada | 7.72e-06 homestead | 0.001  
2355 acre | 0.004942 rood | 0.1977 rod quadrado | 0.1977 perch | 5.98 yd<sup>2</sup> | 5  
3.8 ft<sup>2</sup> | 7750.0 in<sup>2</sup> | 7750015500.0 th<sup>2</sup> |

----- JAPÃO -----

151.25 shaku | 15.125 go | 3.025 jo | 1.5125 tsubo | 1.5125 bu | 0.0504 se |  
0.00504 tan | 0.000504 cho |

----- CHINA 1915#1930 -----

7.5e-05 qing | 0.0075 mu | 0.075 fen | 0.45 fang zhang | 0.75 li - 庫 ou 驪 |  
7.5 hao | 45.0 fang chi | 4500.0 fang cun |

----- CHINA 1930#HOJE -----

8.14e-05 qing | 0.00814 mu | 0.0814 fen | 0.4883 fang zhang | 0.814 li - 庫  
ou 驪 | 8.14 hao | 48.83 fang chi | 4883.0 fang cun |

----- TAILÂNDIA -----

0.003125 rai | 0.0125 ngan | 1.25 tarang wah |

----- GRÉCIA -----

0.005265 plethon | 0.021055 aroura | 0.03158 hektos | 0.06315 hēmiektois |  
0.12185 dodecaorgion | 0.1462 decaorgion | 0.5265 akaina | 1.462 hexapodēs  
ou orgya | 2.1055 bema ou diploum | 52.65 pouss |

----- ROMA -----

2.1245e-06 saltus | 8.5e-06 centurium | 0.00085 geredium | 0.0016995 uger |  
0.0033995 akt quadrado | 0.0136 klima | 0.1224 akt quadrado pequeno | 0.489  
5 decimpeda quadrada | 48.95 ped quadrado comum | 57.0 ped quadrado legal  
|

----- FRANCÊS ANTIGO -----

0.0014625 arpont | 0.14625 perche | 1.316 toise quadrada | 47.385 pied | 6.8  
25 pouce |

----- ESPANHA ANTIGA -----

2.8625e-07 legua | 7.155e-06 labor | 1.294e-05 caballería | 1.553e-05 yugad  
a | 0.0007765 fanegada | 0.001118 aranzada | 0.001272 cuerda de Puerto Rico  
| 0.009315 celemin | 0.03727 cuartillo | 0.44725 estadal quadrado | 7.155 va  
ra quadrada | 64.4 pie quadrado |

----- PORTUGAL ANTIGO -----

1.033 braça quadrada | 4.132 vara quadrada | 103.3 palmo quadrado | 6610.0  
polegada quadrada |

**RÚSSIA ANTIGA**

0.00045765 desyatina país | 0.00034325 desyatina fazendeiros | 4.3935e-06  
verst quadrado | 9.885 arshin quadrado | 58.8 pé quadrado |

**AINDA EM USO**

0.005 dulum (дулум) da Bósnia, Herzegovina e Sérvia | 0.005 dynym ou dylym da Albânia | 0.01223 donum do Chipre | 0.005 skales (σκάλες) do Chipre | 0.005 stremma grega | 0.003937 stremma turca da Grécia | 0.002 dunam do Iraque | 0.005 dunam (dönüm, دونم) da Turquia, Síria, Israel, Palestina, Jordânia, Líbano | 0.00544 dunam (dönüm, دونم) antigo (anterior a 1928) da Turquia, Síria, Israel, Palestina, Jordânia, Líbano | 0.0011905 feddan (فدان) do Egito, Sudão, Síria, Omã | 0.02857 kīrat (قيراط) do Egito | 0.0005 manzana da Argentina | 0.0005985 manzana da Belize | 0.0007155 manzana da Costa Rica | 0.0007085 manzana da Guatemala | 0.000717 manzana da Honduras | 0.0007145 manzana da Nicarágua | 0.0002964 cuadra da Argentina | 0.00031795 cuadra do Chile | 0.0007085 cuadra do Equador | 0.0006665 cuadra do Paraguai | 0.0005 cuadra do Peru | 0.0006775 cuadra do Uruguai | 7.815 vara quadrada do Panamá e da Colômbia | 7.085 vara quadrada do Chile, Costa Rica, República Dominicana, Equador, El Salvador, Guatemala, Honduras, México, Nicarágua, Peru, São Tomé e Príncipe, Venezuela | 6.955 vara quadrada de Cuba | 6.68 vara quadrada da Argentina, Paraguai, Uruguai | 4.132 vara quadrada do Brasil |

**BITOLA DOS FIOS**

9870.0 polegada circular | 9867625.0 kcmil, MCM | 9867626205.0 cmil |

**NATURAIS**

1.914e-68 L<sup>2</sup> |  
A CATEC agradece a sua consulta!

In [ ]: