



UNIVERSIDADE
VILA VELHA
ESPÍRITO SANTO

LABORATÓRIO DE PROGRAMAÇÃO
TRABALHO - REPETIÇÃO

NOME: Nicolás De Paula Muciaccia

MEU LINK COLAB: [Link](#)

3) Criar duas funções e seus protótipos que calculam a Combinação e o Arranjo de n elementos combinados p a p . Após isso, na célula do programa principal, use as funções criadas várias vezes e exiba o resultado da Combinação e do Arranjo enquanto os valores lidos do usuário: n e p estiverem corretos.

Sabe-se que:

$\text{Arranjo}(n, p) = n! / (n-p)!$

$\text{Combinação}(n, p) = \text{Arranjo}(n, p) / p!$

; sendo que: $n \geq 0$ e $n \geq p$

✓ EXERCÍCIO NÚMERO 10:

Crie somente uma (1) função e seu protótipo, conforme para a descrição dos exercícios: 3 acima.

```

1  def combinacaoArranjo(N = -1, P = -1):
2      '''
3      Em casos de erro, caso caia no try, o return = -1
4      '''
5      try:
6          # COMO DEVERIA SER TUDO EM *UMA FUNÇÃO*, ENTÃO INCREMENTEI O FATORIAL NECESSÁRIO AQUI DENTRO, NÃO CRIEI
7          if(N < 0):
8              return 1, 1
9          else:
10             fator1 = 1
11             for i in range(2, N + 1):
12                 fator1 = fator1 * i
13             fator2 = 1
14             for i in range(2, (N - P) + 1):
15                 fator2 = fator2 * i
16             fator3 = 1
17             for i in range(2, P + 1):
18                 fator3 = fator3 * i
19             # FIM DO FATORIAL
20             arran = fator1 / fator2
21             # ARRANJO
22             if(fator3 * fator2 <= 0):
23                 return -1, -1
24             else:
25                 combina = arran / fator3
26                 # COMBINAÇÃO
27                 return arran, combina
28     except:
29         return -1

```

```

1 while True:
2     try:
3         print('INSIRA 1 PARA CALCULAR ARRANJO E COMBINAÇÃO')
4         print('INSIRA 0 PARA ENCERRAR')
5         opcao = int(input('INSIRA A OPÇÃO: '))
6         if(opcao == 0):
7             print('PROGRAMA ENCERRADO!!')
8             break
9         elif(opcao != 1):
10            print('ERRO NOS VALORES DE ENTRADA, TENTE NOVAMENTE!')
11        else:
12            while True:
13                try:
14                    print('INSIRA VALORES DE N >= 0: ', end= ' ')
15                    N = int(input())
16                    if(N < 0):
17                        print('ERRO NOS VALORES DE ENTRADA')
18                    else:
19                        break
20                except:
21                    print('ERRO NOS VALORES DE ENTRADA')
22            while True:
23                try:
24                    print('INSIRA VALORES DE P <= N: ', end= ' ')
25                    P = int(input())
26                    if(P < 0 or (N - P) < 0):
27                        print('ERRO NOS VALORES DE ENTRADA')
28                    else:
29                        break
30                except:
31                    print('ERRO NOS VALORES DE ENTRADA')
32            resultArranjo, resultCombina = combinacaoArranjo(N, P)
33            print(f'ESSE É O RESULTADO DO ARRANJO: {resultArranjo: .1f}')
34            print(f'ESSE É O RESULTADO DA COMBINAÇÃO: {resultCombina: .1f}')
35        except Exception as erro:
36            print(f'ERRO: {erro}')

```

✓ EXERCÍCIO NÚMERO 11:

Crie uma função e seu protótipo que receba como argumento um ano qualquer e um dos três números citados: 4 ou 100 ou 400 e retorne zero ou um (se o ano for ou não bissexto), quociente e resto da divisão do ano pelo número escolhido, conforme as regras de teste do ano bissexto. Após isso, use a função no programa principal e exiba as informações sobre o ano conforme os dados ditos pelo usuário. Regras de teste do ano bissexto:

- ✓ São bissextos todos os anos múltiplos de 400. Exemplo: 1600, 2000, 2400, 2800...
- ✓ São bissextos todos os múltiplos de 4 e não múltiplos de 100. Exemplo: 1996, 2004, 2008, ...
- ✓ Não são bissextos todos os demais anos.

```

1 def anoBissexto(ano, numero):
2     """
3     Em casos de erro, caso caia no try, o return = -1
4     """
5     try:
6         if((ano % 400 == 0) or (ano % 4 == 0 and ano % 100 != 0)):
7             quociente = ano // numero
8             resto = ano % numero
9             return True, quociente, resto
10        else:
11            quociente = ano // numero
12            resto = ano % numero
13            return False, quociente, resto
14    except:
15        return -1

```

```

1 print('CALCULADORA DE ANO BISSEXTO(NÃO É BISSEXTO SE SAÍDA = 0, É BISSEXTO SE SAÍDA = 1)')
2 while True:
3     try:
4         print('DESEJA FAZER O CÁLCULO? 1 P/ SIM ou 0 P/ ENCERRAR O PROGRAMA')
5         opcao = int(input())
6         if(opcao == 0):
7             print('PROGRAMA ENCERRADO!')
8             break
9         elif(opcao != 1):
10            print('ERRO NOS VALORES DE ENTRADA, TENTE NOVAMENTE!')
11        else:
12            while True:
13                try:
14                    print('INSIRA O ANO DESEJADO PARA O CÁLCULO', end= ' ')
15                    ano = int(input())
16                    if(ano <= 0):
17                        print('ERRO NOS VALORES DE ENTRADA')
18                    else:
19                        break
20                except:
21                    print(f'ERRO NOS VALORES DE ENTRADA!')
22            while True:
23                try:
24                    print('ESCOLHA UM NÚMERO (4 ou 100 ou 400)', end= ' ')
25                    numero = int(input())
26                    if(numero != 4 and numero != 100 and numero != 400):
27                        print('ERRO NOS VALORES DE ENTRADA')
28                    else:
29                        break
30                except:
31                    print(f'ERRO NOS VALORES DE ENTRADA!')
32            resposta, quociente, resto = anoBissexto(ano, numero)
33            if(resposta == True):
34                print(f'O ANO {ano} É UM ANO BISSEXTO!!')
35                print(f'O VALOR "{quociente}" É O QUOCIENTE DA DIVISÃO DE {ano}/{numero}')
36                print(f'O VALOR "{resto}" É O RESTO DA DIVISÃO DE {ano}/{numero}')
37            else:
38                print(f'O ANO {ano} NÃO É UM ANO BISSEXTO!!')
39                print(f'O VALOR "{quociente}" É O QUOCIENTE DA DIVISÃO DE {ano}/{numero}')
40                print(f'O VALOR "{resto}" É O RESTO DA DIVISÃO DE {ano}/{numero}')
41        except Exception as erro:
42            print(f'ERRO: {erro}')

```

✓ EXERCÍCIO NÚMERO 12:

12) Crie uma função e seu protótipo que receba como argumento um número inteiro qualquer representando os segundos de realização de um teste de laboratório e retorne as horas, minutos e segundos convertidos deste número. Use a função (100 vezes) no programa principal e exiba a informação, conforme o exemplo a seguir:

Exemplo: Tempo: 10000 Segundos = 2 Horas(s) + 46 Minuto(s) + 40 Segundos(s).

```

1 def conversorTempo(N = 0):
2     '''
3     Em casos de erro, caso caia no try, o return = -1
4     '''
5     try:
6         hora = N // 3600
7         minuto = (N // 60) % 60
8         segundo = N % 60
9         return hora, minuto, segundo
10    except:
11        return -1

```

```

1 print('CALCULADORA QUE CONVERTE SEGUNDOS EM MINUTOS, HORAS e SEGUNDOS')
2 horas, minutos, segundos, quantidade = 0, 0, 0, 0
3 while quantidade < 100:
4     try:
5         print('CASO QUEIRA CALCULAR INSIRA 1, SE NÃO, INSIRA 0: ', end=' ')
6         opcao = int(input())
7         if(opcao == 0):
8             print('PROGRAMA ENCERRADO')
9             break
10        elif(opcao != 1):
11            print('ERRO NOS VALORES DE ENTRADA')
12        else:
13            while True:
14                try:
15                    print('INSIRA O TEMPO EM SEGUNDOS GASTO NO SEU TESTE:', end=' ')
16                    tempo = int(input())
17                    if(tempo < 0):
18                        print('VALORES INVÁLIDOS: INSIRA NOVAMENTE!')
19                    else:
20                        break
21                except:
22                    print('VALORES INVÁLIDOS: INSIRA NOVAMENTE!')
23            quantidade += 1
24            horas, minutos, segundos = conversorTempo(tempo)
25            print(f'Tempo: {tempo} Segundos = {horas} Horas(s) + {minutos} Minuto(s) + {segundos} Segundos(s)')
26        except Exception as erro:
27            print(f'ERRO: {erro}')

```

✓ EXERCÍCIO NÚMERO 13:

Crie uma função e protótipo que receba como argumento um valor real do Tempo T (Horas) e converta em: Horas + Minutos + Segundos. Na célula do programa principal, use a função 50 vezes para valores de T lidos do usuário.

Exemplo: T (12.47 Horas) = 12 Horas(s) + 28 Minuto(s) + 12 Segundo(s).

```

1 def conversorTempoReal(N = 0):
2     '''
3     Em casos de erro, caso caia no try, o return = -1
4     '''
5     try:
6         horas = N // 1
7         minutos = round(N % 1 * 60, 0)
8         segundos = ((N % 1 * 60) % minutos) * 60
9         return horas, minutos, segundos
10    except:
11        return -1

1 print('CONVERSO DE TEMPO EM HORAS, MINUTOS e SEGUNDOS:')
2 contador = 0
3 while contador < 50:
4     try:
5         print('CASO QUEIRA CALCULAR INSIRA 1, SE NÃO, INSIRA 0: ', end=' ')
6         opcao = int(input())
7         if(opcao == 0):
8             print('PROGRAMA ENCERRADO')
9             break
10        elif(opcao != 1):
11            print('ERRO NOS VALORES DE ENTRADA')
12        else:
13            while True:
14                try:
15                    print('INSIRA O TEMPO(em horas):', end='')
16                    tempo = float(input())
17                    if(tempo < 0):
18                        print('ERRO NOS VALORES DE ENTRADA, TENTE NOVAMENTE!!')
19                    else:
20                        break
21                except:
22                    print('ERRO NOS VALORES DE ENTRADA, TENTE NOVAMENTE!!')
23            contador += 1
24            horas, minutos, segundos = conversorTempoReal(tempo)
25            print(f'TEMPO: ({tempo: .2f} Horas) = {horas: .0f} Horas(s) + {minutos: .0f} Minuto(s) + {segundos: .0f} Segundo(s).')
26        except Exception as erro:
27            print(f'ERRO: {erro}')

```

✓ EXERCÍCIO NÚMERO 19:

Faça uma função em Python que receba três valores A, B, C e verifique se os mesmos podem formar um triângulo ou não. Caso possam, dizer que tipo: Escaleno, Isóscele ou Equilátero (0, 2 ou 3 lados iguais, respectivamente).

Condição obrigatória: É triângulo se, somente se $(A < B + C)$ e $(B < A + C)$ e $(C < A + B)$.

Na célula do programa principal, use a função várias vezes, criando um Menu para o usuário.

```
1 def triangulos(A, B, C):
2     """
3     Em casos de erro, caso caia no try, o return = -1
4     """
5     try:
6         if((A < (B + C)) and (B < (A + C)) and (C < A + B)):
7             if(A != B and A != C and B != C):
8                 return "A, B, C FORMAM UM TRIÂNGULO ESCALENO."
9             elif(A == B and A == C and B == C):
10                return "A, B, C FORMAM UM TRIÂNGULO EQUILÁTERO!"
11            else:
12                return "A, B, C FORMAM UM TRIÂNGULO ISÓSCELES"
13        else:
14            return "A,B, C NÃO PODEM FORMAR UM TRIÂNGULO QUALQUER"
15    except:
16        return -1

1 while True:
2     try:
3         print('MENU DE ESCOLHAS, DIGITE: 1 P/ CALCULAR ou 0 P/ ENCERRAR O PROGRAMA ')
4         opcao = int(input('INSIRA A OPÇÃO: '))
5         if(opcao == 0):
6             print('PROGRAMA ENCERRADO!')
7             break
8         elif(opcao != 1):
9             print('ERRO NOS VALORES DE ENTRADA!')
10        else:
11            while True:
12                try:
13                    print('INSIRA O VALOR DO CATETO ("A" > 0): ', end='')
14                    A = float(input())
15                    if(A <= 0):
16                        print('ERRO NOS VALORES DE ENTRADA, TENTE NOVAMENTE!')
17                    else:
18                        break
19                except:
```