



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER  
ESCOLA SUPERIOR POLITÉCNICA  
CST ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - DISTÂNCIA  
LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

**ATIVIDADE PRÁTICA**

FELIPE MARCHI GUIMARÃES – RU: 1371185  
VINICIUS POZZOBON BORIN

RIO CLARO - SP  
2021

## EXERCÍCIO 1

Escreva um programa que leia o nome de um lutador e seu peso. Em seguida, informe a categoria a que pertence o lutador, conforme a Tabela a seguir (note que a tabela foi criada para efeito deste exercício e não condiz com qualquer categoria de luta).

A saída do programa deve exibir na tela uma frase com o padrão descrito a seguir:

Nome fornecido: Pepe Jordão

Peso fornecido: 73.4

Frase a ser exibida: O lutador Pepe Jordão pesa 73,4 kg e se enquadra na categoria Ligeiro

### Solução do aluno:

```
# função que irá ler e retornar o nome digitado pelo usuário
def lerNome(pergunta):
    while True:
        try:
            n = input(pergunta)
            return n
        except:
            print('Algum erro aconteceu.')

# função que irá ler e retornar o peso digitado pelo usuário.
def lerPeso(pergunta):
    while True:
        try:
            p = -1 # peso inicia negativo para entrar no while abaixo, que irá tratar pesos negativos
            while (p < 0):
                p = float(input(pergunta)) # leitura do peso. Se peso for positivo, sairá do while
            if(p < 0):
                print('O valor digitado deve ser positivo.')
        except ValueError:
            print('O dado informado não é um número real.') # trata o valor digitado. O valor precisa ser real (float)
        else:
            return p
```

# função que irá receber um peso por parâmetro e irá retornar a categoria de acordo com o peso informado

```
def categoria(p):  
    cat = ''  
    if (p < 65):  
        cat = 'Pena'  
    elif (p >= 65) and (p < 72):  
        cat = 'Leve'  
    elif (p >= 72) and (p < 79):  
        cat = 'Ligeiro'  
    elif (p >= 79) and (p < 86):  
        cat = 'Meio-Médio'  
    elif (p >= 86) and (p < 93):  
        cat = 'Médio'  
    elif (p >= 93) and (p < 100):  
        cat = 'Meio-Pesado'  
    elif (p >= 100):  
        cat = 'Pesado'  
    return cat
```

# função que irá imprimir na tela o nome, o peso e um texto informando a categoria

```
def impressao (nome, peso, categ):  
    print('-' * 70)  
    print(f'Nome fornecido: {nome}')    print(f'Peso fornecido: {peso}')    print(f'O lutador {nome} pesa {peso} Kg e se enquadra na categoria {categ}')    print('-' * 70 + '\n')
```

# função que irá ler um dado e obrigar o usuário a digitar um dos caracteres S | s | I | i.

# se for S | s irá retornar 0, que irá fazer o while do Main parar de funcionar, encerrando o programa

# se for I | i irá retornar 1, que irá manter o while do Main, fazendo o usuário inserir novos dados

```
def continua(pergunta):  
    while True:
```

```

try:
    op = input(pergunta)
    while (op != 'S') and (op != 's') and (op != 'I') and (op != 'i'):
        print('Opção inválida!')
        op = input(pergunta)
    if (op == 'S') or (op == 's'):
        return 0
    else:
        return 1
except:
    print('Algum erro aconteceu.')

# Main
sair = 1 # sair vale 1 para que o while seja iniciado
while (sair == 1):
    nome = lerNome('Digite o nome do lutador: ')
    peso = lerPeso('Digite o peso do lutador: ')
    impressao(nome, peso, categoria(peso))
    sair = continua('Digite I ou i para inserir dados de outro lutador | Digite S ou s para sair:')
    print('\n')
print('Encerrando o programa...')

```

### Imagem do código funcionando no seu computador:

```

AP - ex001 x
/home/felipe/PycharmProjects/UninterPython/venv/bin/python "/home/felipe/PycharmProjects/UninterPython/AP - ex001.py"
Digite o nome do lutador: Felipe
Digite o peso do lutador: 78.8
-----
Nome fornecido: Felipe
Peso fornecido: 78.8
O lutador Felipe pesa 78.8 Kg e se enquadra na categoria Ligeiro
-----

Digite I ou i para inserir dados de outro lutador | Digite S ou s para sair: I

Digite o nome do lutador: Maguila
Digite o peso do lutador: 101
-----
Nome fornecido: Maguila
Peso fornecido: 101.0
O lutador Maguila pesa 101.0 Kg e se enquadra na categoria Pesado
-----

Digite I ou i para inserir dados de outro lutador | Digite S ou s para sair: s

Encerrando o programa...

Process finished with exit code 0

```

## EXERCÍCIO 2

Escreva um programa que receba como parâmetro de entrada um número inteiro de 5 dígitos no intervalo fechado [10000, 30000] que represente códigos de produtos vendidos em uma loja. Crie uma função para validar os dados de entrada, obrigando o usuário a respeitar o intervalo e o tipo de dado (inteiro).

Crie uma função que calcule e retorne o dígito verificador do código, utilizando a regra de cálculo explicada a seguir. Por exemplo, considere o código 21853, em que cada dígito é multiplicado por um peso começando em 2, os valores obtidos são somados, e do total obtido calcula-se o resto de sua divisão por 7.

Retorne na função o valor do dígito verificador calculado e imprima na tela o código do produto digitado e seu dígito verificador separado por hífen, como: 21853-5 .

Imprima na tela um teste do seu programa utilizando como código os 5 primeiros dígitos do seu RU. Se seu RU tiver menos de 5 dígitos, complete com zeros. Se seu RU cair fora do intervalo especificado, realize o teste mesmo assim.

### Solução do aluno:

```
#função que irá receber um número inteiro e irá verificar se o número digitado está no
#intervalo desejado. O programa irá perguntar o número até que a condição seja con-
templada.
def ler_int(n):
    min = 10000
    max = 30000

    while True:
        try:
            if (n < min) or (n > max):
                while (n < min) or (n > max):
                    n = int(input('Digite um valor inteiro que esteja no intervalo de [10000 -
30000]: '))
        except ValueError:
            print('Valor digitado deve ser um número inteiro! Tente novamente...')
        else:
            return n #Irá retornar o próprio número digitado
```

```

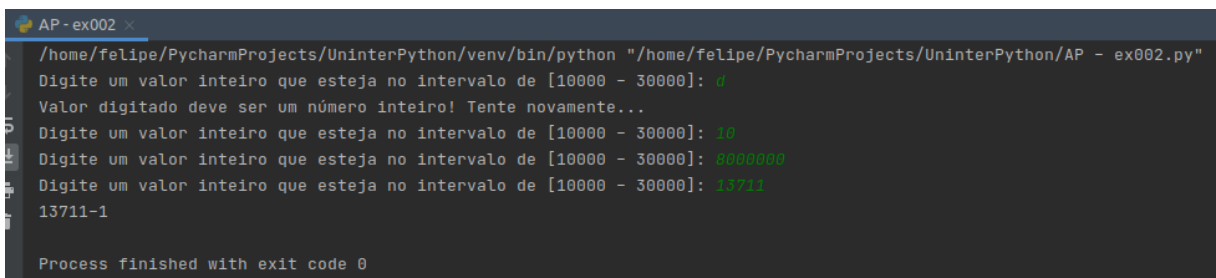
#Função que irá receber o número digitado já validado.
def digito_verificador(n):
    lista_int = []
    lista_peso = [2, 3, 4, 5, 6] #criação de uma lista com o peso para multiplicação
    lista_mult = []
    valor_str = str(n) #variável valor_str irá receber o valor digitado transformado em
string, para que seja possível
    soma = 0          #o acesso de número por número
    for item in valor_str:
        item = int(item) #Para cada item, o próprio item será convertido em inteiro
        lista_int.append(item) #uma lista de inteiro irá receber os inteiros
    for i in range(0, 5, 1):
        lista_mult.append(lista_int[i] * lista_peso[i]) #Esta lista irá receber a multiplicação
do número inteiro e do peso
        soma += lista_mult[i] #variável contém a somatória dos valores da lista com valo-
res multiplicados
    res = (soma % 7)

    return res #retorna o resto da divisão da soma pelo valor 7

# Main
numero = 0
numero = ler_int(numero) #variável número irá receber o valor validado pela função
ler_int()
resto = digito_verificador(numero) #variável resto irá receber o retorno da função digi-
to_verificador()
print(f'{numero}-{resto}')

```

Imagem do código funcionando no seu computador:



```

AP - ex002
/home/felipe/PycharmProjects/UninterPython/venv/bin/python "/home/felipe/PycharmProjects/UninterPython/AP - ex002.py"
Digite um valor inteiro que esteja no intervalo de [10000 - 30000]: 10
Valor digitado deve ser um número inteiro! Tente novamente...
Digite um valor inteiro que esteja no intervalo de [10000 - 30000]: 8000000
Digite um valor inteiro que esteja no intervalo de [10000 - 30000]: 13711
13711-1

Process finished with exit code 0

```

### EXERCÍCIO 3

Considere o seguinte conjunto de dados: Nome + (N1, N2, N3, N4). Nome representa o nome de um aluno e deve ser usado como chave. Já N1, N2, N3, N4 representam as notas de provas desse aluno. Utilize uma estrutura de dicionário com listas para resolver este exercício.

Escreva um programa que leia os dados de N alunos e apresente na tela se foram aprovados ou reprovados. O critério que garante a aprovação é que a média aritmética das 4 notas seja maior ou igual 7,0. O valor de N é a quantidade de alunos, e esse valor deve ser lido do teclado no começo do programa. Faça um laço de repetição para a leitura destes N alunos. As notas devem ser exibidas ao final do programa com uma casa decimal de precisão.

Imprima na tela um teste do seu programa usando como primeiro cadastro o seu nome, e como nota os 4 primeiros dígitos do seu RU.

#### Solução do aluno:

```
# Função que irá retornar a quantidade de alunos a ser registrada
def quantidade_alunos():
    while True:
        try:
            qtd = int(input('Informe a quantidade de alunos que você quer registrar: '))
        except:
            print('x' * 45)
            print('O dado digitado é incorreto! Tente novamente!')
            print('x' * 45)
        else:
            return qtd

# Função que irá realizar o cadastro do aluno e das notas. Ao final do cadastro, haverá
uma lista registro
# contendo um dicionário com chaves Nome, Notas, Média e Conceito, sendo que a cha-
ve Notas contém uma lista.
# Recebe por parâmetro dicionário aluno, lista nota, lista registro, e quantidade de alu-
nos
def cadastro_alunos(aluno, nota, registro, q):
    while True:
        try:
```

```

for cont in range(0, q, 1): # Laço de 0 até a quantidade de alunos-1
    media = 0
    print('-=' * 45)
    aluno['Nome'] = str(
        input(f'Digite o nome do {cont + 1}º aluno(a): ')) # Dicionário na chave
Nome recebe o nome

    nota.append(float(input('N1 = '))) # Lista de notas recebe as 4 notas
    nota.append(float(input('N2 = ')))
    nota.append(float(input('N3 = ')))
    nota.append(float(input('N4 = ')))
    aluno['Notas'] = nota[:] # Lista de notas é copiada para o dicionário na chave
Notas

    for element in nota:
        media += element # soma as 4 notas digitadas na lista
    media /= 4 # calcula a média
    aluno['Média'] = media # Dicionário na chave Média recebe a média calcula-
da

    if (media >= 7): # Verifica o conceito do aluno e o atribui para o dicionário na
chave Conceito
        aluno['Conceito'] = 'Aprovado'
    else:
        aluno['Conceito'] = 'Reprovado'
    registro.append(aluno.copy()) # O dicionario contendo o Nome e Notas é co-
piado para a lista registro

    aluno.clear() # Dicionário é limpo
    nota.clear() # Lista nota é limpa
    print('-=' * 45)

except:
    print('x' * 45)
    print('O dado digitado é incorreto! Tente novamente!')
    print('x' * 45)

else:
    break

# Função que irá imprimir na tela nome do aluno, 4 notas, média e conceito. Recebe por
parâmetro a lista registro
def mostra_alunos(registro):
    print('-' * 80)

```



```

    print(f'{"Notas dos alunos":<30}{"N1":<8}{"N2":<8}{"N3":<8}{"N4":<8}{"Média":<8}{"Situação":<8}')
    print('-' * 80)
    for pos, e in enumerate(registro):
        print(f'e["Nome"]:<30}', end='')
        for i in registro[pos]['Notas']:
            print(f'i:<8.1f}', end='')
        print(f'e["Média"]:<8.1f}{e["Conceito"]:<8}')
    print('-' * 80)

# Main (Parte principal do Programa)

dict_aluno = dict() # dicionário que irá conter as chaves Nome e Notas
list_nota = list() # lista que irá conter 4 notas e armazenada na chave Notas do dicionário
list_registro = list() # lista que irá armazenar o dicionário
qtd = quantidade_alunos()
cadastro_alunos(dict_aluno, list_nota, list_registro, qtd) # Chama a função cadastro_alunos para realizar o cadastro
mostra_alunos(list_registro) # Chama a função mostra_alunos para mostrar os cadastros

```

Imagem do código funcionando no seu computador:

```
AP - ex003 x
/home/felipe/PycharmProjects/UninterPython/venv/bin/python "/home/felipe/PycharmProjects/UninterPython/AP - ex003.py"
Informe a quantidade de alunos que você quer registrar: 4
=====
Digite o nome do 1º aluno(a): Felipe Marchi
N1 = 1
N2 = 3
N3 = 7
N4 = 1
=====
=====
Digite o nome do 2º aluno(a): Cláudio dos Santos
N1 = 6.5
N2 = 3.5
N3 = 5.0
N4 = 6.5
=====
=====
Digite o nome do 3º aluno(a): Eduardo Falco
N1 = 7.5
N2 = 6.5
N3 = 7.0
N4 = 6.5
=====
=====
Digite o nome do 4º aluno(a): Lígia de Oliveira
N1 = 8.5
N2 = 7.5
N3 = 8.0
N4 = 7.0
=====
=====
Notas dos alunos      N1      N2      N3      N4      Média  Situação
-----
Felipe Marchi         1.0     3.0     7.0     1.0     3.0    Reprovado
Cláudio dos Santos    6.5     3.5     5.0     6.5     5.4    Reprovado
Eduardo Falco         7.5     6.5     7.0     6.5     6.9    Reprovado
Lígia de Oliveira     8.5     7.5     8.0     7.0     7.8    Aprovado
=====

Process finished with exit code 0
```

## EXERCÍCIO 4

Leia e armazene em um dicionário o nome, a idade e o número do telefone de seus contatos, sendo que a chave deve ser o nome. Ao digitar uma string vazia para o nome, o programa interrompe a leitura e se encerra.

Apresente na tela os dados lidos em ordem alfabética pelo nome dos contatos. Uma possível solução de ordenar alfabeticamente é usar o método sort.

Em seguida, armazene os contatos em outros dois dicionários, utilizando como critério a idade: menores de 18 anos em um e os maiores em outro dicionário, eliminando o original. Apresente na tela os dois dicionários resultantes da separação.

Imprima na tela um teste do seu programa usando como primeiro cadastro o seu nome, como telefone o seu RU, e como idade os dois últimos dígitos do seu RU.

### Solução do aluno:

```
# importa itemgetter para poder ordenar o dicionário
from operator import itemgetter

# Função que recebe por parâmetro um dicionário e uma lista. Irá realizar o registros do usuário
def inserir_contato(contato, dados):
    while True:
        try:
            op = 'S'
            while (op == 'S'):
                print('(Nome vazio encerra a agenda)')
                nome = str(input('Nome: '))
                if (nome == ""):
                    break # recebe um nome do usuário, se o nome for vazio, o laço é encerrado
            else:
                idade = int(input('Idade: ')) # recebe uma idade do usuário
                tel = str(input('Telefone: ')) # recebe um telefone do usuário
                dados.append(idade) # a lista dados recebe um append da idade e do telefone
                dados.append(tel)
                contato[nome] = dados.copy() # a lista contendo a idade e o telefone é copiada para o dicionário contato
                dados.clear() # a lista é limpa para futuras inserções
                print()
                op = input('Deseja Continuar? [S / N]: ')
                op = op.upper()
                while (op != 'S') and (op != 'N'): # o usuário decide se quer continuar ou não a inserção de dados
                    print()
                    op = input('Deseja Continuar? [S / N]: ')
                    op = op.upper()
                print('-' * 45)
        except ValueError:
            print('x' * 45)
```

```
print('Valor digitado incorreto. Tente novamente!') # trata os dados recebidos pelo usuário
    print('x' * 45)
else:
    print('x' * 90)
    print('\t\t\t\t\tFinalizando agenda...')
    print('x' * 90)
    break

# Função que recebe um dicionário e o ordena em ordem alfabética, tendo como retorno uma lista ordenada
def mostrar_contato(contato):
    ordered_dict = list()
    ordered_dict = sorted(contato.items(), key=itemgetter(0)) # uma lista recebe a ordenação do dicionário
    print('-=' * 45)
    print('-' * 60)
    print('<=== AGENDA EM ORDEM ALFABÉTICA ===>\n')
    for pos, item in enumerate(ordered_dict): # imprime a agenda em ordem alfabética
        print(f'Contato {pos+1}:\n')
        print(f'Nome: {item[0]}')
        print(f'Idade: ', item[1][0])
        print(f'Telefone: ', item[1][1])
        print('-' * 60)
    contato.clear() # limpa o dicionário principal, conforme solicitado no enunciado
    return ordered_dict # retorna uma lista ordenada por nome

# Função que recebe por parâmetro a lista orنادada por nome. Dois dicionários novos são criados,
# um recebe dados de menores de idade, o outro, de maiores. Os dois dicionários ordenados por nome
# são mostrados na tela
def lista_idade(dic):
    dict_maior = dict()
    dict_menor = dict()
    list_dados = list()

    for item in dic:
        if (item[1][0] >= 18): # verifica na lista se a idade é maior ou igual a 18
            list_dados.append(item[1][0])
            list_dados.append(item[1][1]) # uma nova lista recebe a idade e o telefone
            dict_maior[item[0]] = list_dados.copy() # o dicionário de maiores de idade, na chave Nome, recebe a lista com idade e telefone
        else:
            list_dados.append(item[1][0])
            list_dados.append(item[1][1])
            dict_menor[item[0]] = list_dados.copy() # faz igual ao if anterior, mas para menores de idade
    list_dados.clear() # limpa a lista a cada iteração do for
    print('-=' * 45)
    print('-' * 60)
    print('<=== MENORES DE IDADE ===>\n')
    for key, item in dict_menor.items(): # mostra na tela os menores de idade
        print(f'Nome: {key}')
        print(f'Idade: {item[0]}')
        print(f'Telefone: {item[1]}')
        print('-' * 45)
```

```
print('-=' * 45)
print('-' * 60)
print('<== MAIORES DE IDADE ==>\n')
for key, item in dict_maior.items(): # mostra na tela os maiores de idade
    print(f'Nome: {key}')
    print(f'Idade: {item[0]}')
    print(f'Telefone: {item[1]}')
    print('-' * 45)
print('-=' * 45)

# Main (Parte principal do programa)

dict_contato = dict()
list_dados = list()

inserir_contato(dict_contato, list_dados) # chama a função inserir_contato para o usuário
realizar o cadastro
ord_dict = mostrar_contato(dict_contato) # lista que recebe a lista ordenada da função
mostrar_contato()
lista_idade(ord_dict) # chama a função lista_idade para mostrar os dicionários por idade
```

Imagem do código funcionando no seu computador:

```
AP - ex004 ×
/home/felipe/PycharmProjects/UninterPython/venv
(Nome vazio encerra a agenda)
Nome: Felipe
Idade: 13
Telefone: 1371185

Deseja Continuar? [S / N]: s
-----
(Nome vazio encerra a agenda)
Nome: Fabio
Idade: 18
Telefone: 222222

Deseja Continuar? [S / N]: s
-----
(Nome vazio encerra a agenda)
Nome: Alvaro
Idade: 33
Telefone: 333333

Deseja Continuar? [S / N]: s
-----
(Nome vazio encerra a agenda)
Nome: Papa Capim
Idade: 40
Telefone: 444444

Deseja Continuar? [S / N]: s
-----
(Nome vazio encerra a agenda)
Nome: Evaristo
Idade: 15
Telefone: 55555

Deseja Continuar? [S / N]: s
-----
(Nome vazio encerra a agenda)
Nome: Gaúcho
Idade: 39
Telefone: 66666

Deseja Continuar? [S / N]: n
-----
```

```
-----
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                          Finalizando agenda...
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----
-----
<=== AGENDA EM ORDEM ALFABÉTICA ===>

Contato 1:

Nome: Alvaro
Idade: 33
Telefone: 333333
-----

Contato 2:

Nome: Evaristo
Idade: 15
Telefone: 55555
-----

Contato 3:

Nome: Fabio
Idade: 18
Telefone: 22222
-----
```

```
-----

Contato 4:

Nome: Felipe
Idade: 13
Telefone: 1371185
-----

Contato 5:

Nome: Gaúcho
Idade: 39
Telefone: 66666
-----

Contato 6:

Nome: Papa Capim
Idade: 40
Telefone: 444444
-----
```

```
-----  
-----  
<=== MENORES DE IDADE ===>  
  
Nome: Evaristo  
Idade: 15  
Telefone: 55555  
-----  
  
Nome: Felipe  
Idade: 13  
Telefone: 1371185  
-----
```

```
-----  
-----  
<=== MAIORES DE IDADE ===>  
  
Nome: Alvaro  
Idade: 33  
Telefone: 333333  
-----  
  
Nome: Fabio  
Idade: 18  
Telefone: 222222  
-----  
  
Nome: Gaúcho  
Idade: 39  
Telefone: 66666  
-----  
  
Nome: Papa Capim  
Idade: 40  
Telefone: 444444  
-----  
-----  
  
Process finished with exit code 0
```