



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER  
ESCOLA SUPERIOR POLITÉCNICA  
TECNÓLOGO EM CIÊNCIA DE DADOS  
LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

**ATIVIDADE PRÁTICA**

IANN OLIVEIRA SILVA – RU: 3698289  
PROF. VINICIUS POZZOBON BORIN

MUCAJAÍ – RORAIMA  
2021

## EXERCÍCIO 1:

Escreva um programa que leia o nome de um aluno e sua nota final. Em seguida, informe o conceito conforme a tabela abaixo. A saída do programa deve exibir na tela uma frase com o padrão descrito a seguir:

*Nome do aluno: Fábio José*

*Nota final: 3.5*

*Frase a ser exibida: O aluno Fábio José tirou nota 3.5 e se enquadra no conceito D*

Nota	Conceito
De 0,0 a 2,9	E
De 3,0 a 4,9	D
De 5,0 a 6,9	C
De 7,0 a 8,9	B
De 9,0 a 10	A

Todos os dados devem ser lidos do teclado, sendo que o nome do aluno é uma string e a nota final é um número real. Não é necessário armazenar os dados em uma estrutura de dados, basta imprimir na tela. Coloque todo o seu programa dentro de um laço de repetição e faça-o se encerrar quando uma determinada condição for satisfeita. A condição fica a seu critério, como por exemplo, encerrar o programa ao digitar a palavra sair, ou então uma nota inválida.

Imprima na tela um teste do seu programa utilizando o seu nome e os dois últimos dígitos do seu RU para a nota.

### Solução:

```
# Cabeçalho do programa
print('-' * 50)
print('----- Exercício 1 da Atividade Prática -----')
print('-' * 50)

# Função para validar a entrada do usuário, uma entrada diferente da esperada, irá retornar uma mensagem e solicitará
```

# uma nova entrada. Peguei como base o algoritmo de validação apresentado na Aula 5

```
def entrada():
```

```
    entrada = int(input('Selecione uma opção: '))
    while (entrada < 1) or (entrada > 2):
        print('Opção inválida... Tente novamente ')
        entrada = int(input('Selecione uma opção: '))
        print('-' * 50)
    return entrada
```

# Função que irá receber o nome do aluno e sua nota. O strip() irá apagar espaços desnecessários na entrada do usuário.  
# No IF o programa irá comparar a Nota Final do aluno com as condições apresentadas, se ela não for suprida, passará  
# para a próxima condição. Se a condição for suprida, mostrará na tela o print com a mensagem informando o Nome, Nota e  
# Conceito do aluno. Se o valor da nota inserido for maior que 10 o ELSE irá mostrar uma mensagem de erro e solicitará  
# novos dados ao usuário, chamando novamente a função func\_notas().

```
def func_notas():
```

```
    nome = str(input('Nome do aluno: ')).strip()
    nota = float(input('Nota do final: '))
    if 9 <= nota <= 10:
        print('-' * 50)
        print('Nome do aluno: {}\n'
              'Nota final: {}\n'
              'O aluno {} tirou nota {} e se enquadra no conceito A.'.format(nome, nota, nome, nota))
        print('-' * 50)
    elif 7 <= nota <= 8.9:
        print('-' * 50)
        print('Nome do aluno: {}\n'
              'Nota final: {}\n'
              'O aluno {} tirou nota {} e se enquadra no conceito B.'.format(nome, nota, nome, nota))
        print('-' * 50)
    elif 5 <= nota <= 6.9:
        print('-' * 50)
        print('Nome do aluno: {}\n'
              'Nota final: {}\n'
              'O aluno {} tirou nota {} e se enquadra no conceito C.'.format(nome, nota, nome, nota))
        print('-' * 50)
    elif 3 <= nota <= 4.9:
        print('-' * 50)
```

```

    print('Nome do aluno: {}\n'
          'Nota final: {}\n'
          'O aluno {} tirou nota {} e se enquadra no conceito D.'.format(nome, nota, nome, nota))
    print('-' * 50)
elif 0 <= nota <= 2.9:
    print('-' * 50)
    print('Nome do aluno: {}\n'
          'Nota final: {}\n'
          'O aluno {} tirou nota {} e se enquadra no conceito E.'.format(nome, nota, nome, nota))
    print('-' * 50)
else:
    print('-' * 50)
    print('Valor da nota excede o limite. Insira os dados novamente.')
    print('-' * 50)
    func_notas()

# Laço de repetição que manterá o usuário no programa até que ele deseje sair, selecionando a opção
# '2 - Sair do programa'. Enquanto o laço for verdadeiro 'True', ele continuará sendo executado. A variável 'menu'
# receberá 1 ou 2, que foi recebido pela função entrada(), e acionará uma opção do menu do programa. Algo diferente
# não é aceito. Se o valor recebido pela variável menu, for igual a 1, o IF chamará a função func_notas().
# Se a entrada for igual a 2, o programa irá encerrar.
while True:
    print(' 1 - Inserir dados.')
    print(' 2 - Sair do programa.')
    print('-' * 50)
    menu = entrada()
    if menu == 1:
        func_notas()
    else:
        break

```

Imagem do código funcionando:

```
----- Exercício 1 da Atividade Prática -----  
-----  
1 - Inserir dados.  
2 - Sair do programa.  
-----  
Selecione uma opção: 1  
Nome do aluno: Iann  
Nota do final: 8.9  
-----  
Nome do aluno: Iann  
Nota final: 8.9  
O aluno Iann tirou nota 8.9 e se enquadra no conceito B.  
-----  
1 - Inserir dados.  
2 - Sair do programa.  
-----  
Selecione uma opção: 2  
-----  
Process finished with exit code 0
```

## EXERCÍCIO 2:

Faça uma função que receba o nome e sobrenome de uma pessoa e retorne a primeira letra de seu nome e seu sobrenome concatenando com a string @algoritmos.com.br. No algoritmo principal deverá ser apresentada a mensagem ao usuário contendo seu nome completo e seu email. Imprima na tela um teste do seu programa utilizando o seu nome e sobrenome concatenado com os dois últimos dígitos do seu RU.

Sra Luciane Kanashiro, seu email é [kanashiro16@algoritmos.com.br](mailto:kanashiro16@algoritmos.com.br)

Solução:

```

# Primeiro foi declarado uma variável para a string '@algoritmos.com.br', logo em seguida é pedido ao usuário que
# insira seu nome, note que foram chamadas três funções, lower() que irá transformar toda o input do usuário
# em minúsculo caso ele entre com algum caractere maiúsculo, na função strip() será eliminado todos os espaços
# desnecessários, caso ela não fosse declarada o programa não funcionaria da maneira correta, a função split()
# irá retornar uma lista da string, como não foi declarado um separador, ela separará a string sempre que um espaço ' '
# for identificado, exemplo, pegando como input 'lann Oliveira Silva' o output seria ['lann', 'Oliveira', 'Silva'],
# e por ultimo é solicitado ao usuário que informe seu RU.
def email_aluno():
    nomeReservado = str() # Uma variável string que declarei e deixei vazia, para utilizar como reserva do nome do
    # usuário.
    padrao = '@algoritmos.com.br'
    nome = input('Digite seu nome e sobrenome: ').lower().strip().split()
    ru = input('Digite seu RU: ')
    aluno = nome[0][0] + nome[1] # aluno receberá como valor a primeira letra da primeira string da lista nome[0][0] e
    # fará a concatenação com a segunda string da lista na posição nome[1].
    posiRU = ru[len(ru) - 2] # Nesse momento len irá fazer a contagem da variável RU, encontrado o tamanho da variável,
    # logo após é subtraído 2, encontrando assim a penúltima posição do RU do aluno, posiRU recebe o valor
    # referente a penúltima posição.
    posiRU_dois = ru[len(ru) - 1] # Aqui é encontrado a última posição do RU e o seu valor é atribuído a posiRU_dois.
    email = aluno + posiRU + posiRU_dois + padrao # Nessa linha será feita apenas a concatenação de todos os elementos
    # encontrados aluno+posiRU+posiRU_dois+padrao.
    for i in range(len(nome)): # Fiz um FOR que usará o tamanho da lista nome como range, usando a função len. A cada
    # instância do FOR nomeReservado receberá o valor de nome na posição i e logo em seguida acrescentara um
    # espaço ' ' entre os valores recebidos pois foram recebidos como string e para evitar que fiquem colados,
    # utilizei esse artifício.
    nomeReservado += nome[i]
    nomeReservado += ' '
    nomePrint = nomeReservado.strip().title() # Depois do for, eu atribui como valor para a variável nomePrint, a
    # variável nomeReservado chamando as funções strip() para eliminar o ultimo espaço ' ' e a função title(), que irá
    # retornar maiúsculo os primeiros caracteres de cada string.
    print('Sr(a) {}, seu email é {}'.format(nomePrint, email)) # Esse é o ultimo print, que usando a função format
    # e utilizando como parâmetros as variáveis nomePrint e email para reproduzir na tela a frase completa do exercício.

email_aluno() # Linha que somente irá chamar a função elaborada.

```

Imagem do código funcionando:

```
/home/iann/PycharmProjects/AtividadePratica/venv/bin/python /home/iann/
Digite seu nome e sobrenome: iann oliveira silva
Digite seu RU: 3698289
Sr(a) Iann Oliveira Silva, seu email é ioliveira89@algoritmos.com.br

Process finished with exit code 0
```

### EXERCÍCIO 3:

Um canal de jogos do youtube está fazendo um sorteio para angariar doações para pessoas em situação de vulnerabilidade social. A cada 10,00 doado o nome da pessoa é inserido em uma lista de sorteio, por exemplo:

Ruth doou 20,00;

Maria doou 30,00;

Fernando doou 50,00;

A lista de sorteio estará com os valores:

```
listaSorteio = ['Ruth', 'Ruth', 'Maria', 'Maria', 'Maria', 'Fernando', 'Fernando',
'Fernando', 'Fernando', 'Fernando']
```

Implemente um programa para cadastrar o nome das pessoas que doaram. O programa deve embaralhar a lista, sortear o ganhador e imprimir o seu nome. Imprima na tela um teste do seu programa utilizando como primeiro doador o seu nome e os dois últimos dígitos do seu RU para o valor doado. Não se esqueça de imprimir também a lista de sorteio.

### Solução:

```
from random import shuffle, choice # Importei da biblioteca random duas funções para auxiliar no sorteio do vencedor,
# shuffle que irá embaralhar a lista e a função choice irá sortear um item da lista.
from time import sleep # Da biblioteca time, chamei a função sleep() que faz pausas no código, e me ajudou a
# personalizar um pouco o programa.
from sys import exit # A função exit da biblioteca sys, foi uma das formas que encontrei para finalizar o código
# independente do que viria depois, se essa função não fosse chamada o 'while True', continuaria a rodar, o que não
# faria muito sentido, pois o sorteio teria terminado, e o vencedor declarado.

def menu_adm(): # Nessa função eu elaboro um menu para o administrador do sorteio, onde so ele através de senha tem
    # acesso.
    print('1 - Embaralhar lista')
    print('2 - Verificar lista embaralhada')
    print('3 - Sortear ganhador')
    print('4 - Voltar')
    print('_*' * 40)

select = entrada() # Select vai receber um valor válido para o menu, valor esse recebido pela função entrada() que
# recebe um valor do usuário e verifica se ele é válido ou não.
if select == 1: # Se select receber 1, eu irei fazer um shuffle na listaSorteio e retornarei para a função
    # menu_adm().
    shuffle(listaSorteio)
    print('_*' * 40)
    return menu_adm()
elif select == 2: # Fiz um elif que caso select for igual a 2, faço somente o print da lista listaSorteio e
    # retorno para a função menu_adm().
    print(listaSorteio)
    return menu_adm()
elif select == 3: # Nesse elif será feito o sorteio do vencedor, é declarado uma lista X = ['.'] e printara
    # 'SORTEANDO', logo em seguida.
    x = ['.']
```



```

print('SORTEANDO ')
for i in range(0, 3): # Esse FOR irá fazer uma pequena personalização na saída, chamei a função sleep() e
    # declarei como parâmetro 1, esse 1 corresponde ao total de segundos que o programa ficará pausado.
    # o FOR funcionará assim, quando ele passar o primeiro valor de range, o código irá pausar por 1s e printara
    # na tela um '.' valor correspondente a x[0], como foi colocado um end="" no print, a quebra de linha não
    # ocorrerá e o for se manterá na mesma linha. No fim o FOR terminará assim.
    # sleep(1)'.sleep(1)'.sleep(1)'...'.
    sleep(1)
    print(x[0], end="")
sleep(1) # Pausei o código por 1s novamente.
print(' O sorteado foi... ', end="") # Print um cabeçalho sem a quebra de linha.
sleep(1) # Programa pausado novamente.
print('{}!!! PARABÉNS!'.format(choice(listaSorteio))) # E faço o print do vencedor através do format que recebe
# como valor,, o nome do sorteio que foi sorteado através da função choice.
sleep(1) # Outra pausa de 1s.
print('_*' * 40)
print('_*' * 7, 'A EQUIPE DO CANAL AGRADECE A CONTRIBUIÇÃO DE TODOS', '_*' * 7) # Aqui é printado um cabeçalho,
# agradecendo a todos pela contribuição
print('_*' * 40)
exit() # Logo após o cabeçalho eu chamo a função exit() da biblioteca sys, que irá finalizando o programa e
# e evitando que ele volte para o while True.
elif select == 4:
    return 0 # Caso o usuário insira 4, chamará esse elif e ele retornará 0 para a função menu_adm() e o programa
# retornará para o while True, onde está o programa principal.

def adm(): # Essa função irá servir para que só os administradores do programa tenha acesso a função menu_adm().
    # funcionando da seguinte forma, uma senha predefinida é declarada 123. O usuário ao selecionar a opção
    # '3 - Preparar sorteio', o programa pedirá a ele que insira a senha, no IF é feito a comparação da senha entrada
    # pelo usuário e a senha predefinida, caso eles coincidirem será printado um cabeçalho informando acesso
    # liberado e logo em seguida será feito a chamada da função menu_adm(). Caso a senha informada pelo usuário não
    # for igual ele será jogado no ELSE, que retornará para o usuário a mensagem 'Senha incorreta', e ele será jogado
    # no programa principal.
    senha = 123
    print('Digite a senha para ter acesso: ')
    senhaUsu = int(input())
    if senha == senhaUsu:
        print('_*' * 40)

```

```
print(' ' * 15, 'ACESSO LIBERADO')
print('_*' * 40)
menu_adm()
else:
    print('Senha incorreta')
```

**def entrada():** # Essa função fará a validação da entrada do usuário.

**entrada = int(input())** # O usuário entra com um valor inteiro, que servirá como parâmetro para a seleção de opções  
# no menu do programa principal e o menu da função menu\_adm().

**while entrada > 4 or entrada < 1:** # Nesse while é verificado se o valor inserido pelo usuário está fora do intervalo  
# aceito que seria (1, 4). Caso esteja será printado na tela 'Entrada inválida, tente novamente!' e será  
# solicitado uma nova entrada, e fazendo a comparação novamente. Caso o valor estiver dentro do intervalo  
# definido a função entrada() receberá esse valor.

**print('Entrada inválida, tente novamente!')**

**entrada = int(input())**

**print('--' \* 40)**

**return entrada**

**def main():** # A função main() irá trabalhar da seguinte forma, foram declaradas duas variáveis uma recebe o

# nome da pessoa a outra o valor doado. A variável nomeDoador ao receber o nome, irá transformar todos os caracteres

# para minúsculo com a função lower(), em seguida será excluído todos os espaços desnecessários da string com o

# a função strip(), para fins de formatação foi chamada a função title() que irá retornar as strings com o primeiro

# caractere maiúsculo, e por último é retornado uma lista da string recebida por nomeDoador com a função split().

**nomeDoador = str(input('Digite seu nome: ')).lower().strip().title().split()**

**valorDoador = float(input('Quanto você deseja doar: '))**

**print('\_\*' \* 40)**

**qtdvezes = int(valorDoador / 10)** # A variável qtdvezes irá receber a quantidade de vezes que o nome do doador será

# adicionado a lista, conforme o valor doado, que é dado pelo valor doado dividido por 10.

**for i in range(qtdvezes):** # Nesse FOR é feito a adição do nome do doador na lista do sorteio, o range do FOR é

# definido através da variável que recebe a quantidade de vezes que o nome do doador será adicionado a lista,

# fazendo assim um append do primeiro nome do doador caso ele tenha colocado nome e sobrenome na lista do

# sorteio.

**listaSorteio.append(nomeDoador[0])**

**print('Obrigado por contribuir!!')** # Logo após a adição do nome na lista é printado um cabeçalho agradecendo

# pela contribuição e mostrando também a quantidade de vezes que o nome dele foi adicionado a lista do sorteio.

**print('Seu nome foi adicionado {} vezes na lista para o sorteio.'.format(qtdvezes))**

```

print('_*' * 40)

listaSorteio = ['Ruth', 'Ruth', 'Maria', 'Maria', 'Maria', 'Fernando', 'Fernando', 'Fernando',
               'Fernando', 'Fernando']
# Declarei lista , que ira receber o nome dos doadores, para efetuar o sorteio.

while True: # Para elaboração do menu principal do programa eu utilizei um while True, que ficará sempre executando,
            # caso alguma forma de interrupção não for feita.
            print('SELECIONE UMA OPÇÃO ABAIXO: ') # Nessa e nas 5 linhas abaixo, é feito o menu principal para o usuário,
            # informando o número de cada opção.
            print('1 - Fazer doação')
            print('2 - Ver lista de sorteio')
            print('3 - Preparar para sorteio')
            print('4 - Sair')
            print('_*' * 40)
            select = entrada() # A variável select irá receber o valor da função entrada(), que expliquei anteriormente como é
            # dado esse valor, correspondente a alguma opção do menu, entre (1, 4).
            if select == 1: # Somente é feita a chamada para a função main(), caso o valor do select for igual a 1.
                main()
            elif select == 2: # Caso o valor do select for igual a 2, será printado um cabeçalho e a lista contendo os
                # participantes do sorteio.
                print('_*' * 40)
                print(listaSorteio)
                print('_*' * 40)
            elif select == 3: # Se select for igual a 3, o usuário será direcionado a função adm(), onde será solicitado a ele
                # a senha de acesso para poder acessar o menu do administrador do sorteio. Senha = 123.
                adm()
            elif select == 4: # Se o usuário resolver sair do programa, será mostrado pra ele um agradecimento da equipe
                # do canal e logo em seguida é feito um break no loop while True.
                print('_*' * 40)
                print('_*' * 7, 'A EQUIPE DO CANAL AGRADECE A CONTRIBUIÇÃO DE TODOS', '_*' * 7)
                print('_*' * 40)
                break

```

## Imagens do código funcionando:

```
EX3 x
/home/iann/PycharmProjects/AtividadePratica/venv/bin/python /home/iann/PycharmProjec
SELECIONE UMA OPÇÃO ABAIXO:
1 - Fazer doação
2 - Ver lista de sorteio
3 - Preparar para sorteio
4 - Sair
*****
1
Digite seu nome: iann oliveira silva
Quanto você deseja doar: 89
*****
Obrigado por contribuir!!
Seu nome foi adicionado 8 vezes na lista para o sorteio.
*****
SELECIONE UMA OPÇÃO ABAIXO:
1 - Fazer doação
2 - Ver lista de sorteio
3 - Preparar para sorteio
4 - Sair
*****
|
```

```
*****
SELECIONE UMA OPÇÃO ABAIXO:
1 - Fazer doação
2 - Ver lista de sorteio
3 - Preparar para sorteio
4 - Sair
*****
2
*****
['Ruth', 'Ruth', 'Maria', 'Maria', 'Maria', 'Fernando', 'Fernando', 'Fernando', 'Fernando', 'Fernando', 'Iann', 'Iann', 'Iann', 'Iann', 'Iann',
 'Iann', 'Iann', 'Iann']
*****
SELECIONE UMA OPÇÃO ABAIXO:
```

```

*****
SELECIONE UMA OPÇÃO ABAIXO:
1 - Fazer doação
2 - Ver lista de sorteio
3 - Preparar para sorteio
4 - Sair
*****
3
Digite a senha para ter acesso:
123
*****
                        ACESSO LIBERADO
*****

```

```

1 - Embaralhar lista
2 - Verificar lista embaralhada
3 - Sortear ganhador
4 - Voltar
*****
1
*****
1 - Embaralhar lista
2 - Verificar lista embaralhada
3 - Sortear ganhador
4 - Voltar
*****
2
['Iann', 'Maria', 'Iann', 'Ruth', 'Maria', 'Iann', 'Maria', 'Iann', 'Iann', 'Iann', 'Fernando', 'Fernando', 'Fernando', 'Iann', 'Fernando', 'Iann',
'Ruth', 'Fernando']

```

```

1 - Embaralhar lista
2 - Verificar lista embaralhada
3 - Sortear ganhador
4 - Voltar

*****
3
SORTEANDO
... O sorteado foi... Fernando!!! PARABÉNS!
*****
***** A EQUIPE DO CANAL AGRADECE A CONTRIBUIÇÃO DE TODOS *****
*****
Process finished with exit code 0

```

#### EXERCÍCIO 4:

Considere a tabela a seguir referente a produtos armazenados em um depósito, em que são considerados o estoque atual de cada produto e o estoque mínimo necessário.

Código	Estoque	Mínimo
1	35	20
5	75	50
2	43	45
3	26	18
20	35	20

Armazene as informações acima em uma estrutura de lista com dicionário, substituindo a primeira linha com os dados: no campo código coloque o primeiro dígito do seu RU, no estoque os dois dígitos seguintes de seu RU, e no campo mínimo os dois últimos dígitos do seu RU.

Por exemplo, tendo o RU: 123456

Código	Estoque	Mínimo
1	23	56
5	75	50
2	43	45
3	26	18
20	35	20

As informações devem ser inseridas no dicionário via teclado. Ao digitar o código 0 (zero), o programa interrompe a leitura e se encerra. Ordene os produtos em ordem crescente de código. Imprima na tela um teste do seu programa usando como primeiro cadastro o primeiro dígito do seu RU, como estoque os dois dígitos seguintes de seu RU, e como mínimo os dois últimos dígitos do seu RU.

Dica: A sua lista deve conter o dicionário da seguinte forma:

```
[{'codigo': 1, 'estoque': 23, 'minimo': 56}, {'codigo': 5, 'estoque': 75, 'minimo': 50}, ..., {'codigo': 20, 'estoque': 35, 'minimo': 20}]
```

Para criar dicionário e lista:

```
dicionario = { }
```

```
lista = [ ]
```

*Para ordenar: utilize o método sorted, ordenando pela chave codigo*

```
listaOrdenada = sorted(lista, key=itemgetter('codigo'))
```

*Não se esqueça de importar o itemgetter no início do seu programa.*

```
from operator import itemgetter
```

*Ou utilize o lambda:*

```
listaOrdenada = sorted(lista, key=lambda item: item['codigo'])
```

### Solução:

```
from operator import itemgetter # É feito a importação do método itemgetter da biblioteca operator, método esse que
# servirá para ordenar a lista com base no value da key codigo.

listaTemporaria = [] # Declarei uma lista vazia, que irá receber o dicionário do usuário contendo as keys e values na
# sequência das entradas.
print('-*' * 42)
print('-'*10, 'CADASTRO DE PRODUTOS E CONTROLE DE ESTOQUE', '-'*10) # Apenas um cabeçalho para o programa ficar mais
# intuitivo
print('-'*10, ' Entre com código 0 para sair do programa ', '-'*10)
while True: # Parte principal do programa, onde eu coloquei em um loop com while True, que o usuário somente sairá do
# loop caso ele entre com o valor 0 para a variável código, caso contrário será sempre solicitado a ele que informe
# um valor para a variável código e os values para as keys estoque e minimo.
    print('-*' * 42)
    codigo = int(input('Digite o código do produto: ')) # Variável código que receberá algum valor do usuário, valor
# esse que será adicionado a key 'codigo' no dicionário caso a condição não for suprida, caso a entrada do usuário
# for barrada pelo IF o programa será interrompido pelo comando BREAK.
    if codigo == 0:
        break
    leituraDicio = {'codigo': codigo, # leituraDicio será o dicionário que receberá os values pelo usuário para a keys
# predefinidas, exceto a key código que receberá como value o valor da variável código, inserida e
# testada anteriormente pelo IF.
        'estoque': int(input('Digite a quantidade em estoque: ')),
        'minimo': int(input('Informe a quantidade mínima no estoque: '))}

    listaTemporaria.append(leituraDicio) # Nesta linha do programa, chamei a listaTemporaria[] que até o momento estava
# vazia e chamei também a função append que irá adicionar no final dessa lista um novo dicionário a cada passagem do
# while.
print('Lista: \n', listaTemporaria) # Apenas dei um print da lista com os valores na sequência que o usuário inseriu
# para que seja feita uma comparação com a lista ordenada.
listaOrdenada = sorted(listaTemporaria, key=itemgetter('codigo')) # Declarei uma variável que chama o método SORTED,
# método esse que irá ordenar a listaTemporaria, usando como parâmetro o value das keys código, values que a
# função itemgetter chamou quando foi declarado a key('codigo') como parâmetro de itemgetter.
print('lista Ordenada: \n', listaOrdenada) # Print da lista ordenada de forma crescente, tendo como base os values das
# keys 'codigo'.
```



## Imagens do código funcionando:

```
*****
***** CADASTRO DE PRODUTOS E CONTROLE DE ESTOQUE *****
***** Entre com código 0 para sair do programa *****
*****
Digite o codigo do produto: 3
Digite a quantidade em estoque: 69
Informe a quantidade mínima no estoque: 89
*****
Digite o codigo do produto: 5
Digite a quantidade em estoque: 75
Informe a quantidade mínima no estoque: 50
*****
Digite o codigo do produto: 2
Digite a quantidade em estoque: 43
Informe a quantidade mínima no estoque: 45
*****
Digite o codigo do produto: 3
Digite a quantidade em estoque: 26
Informe a quantidade mínima no estoque: 18
*****
Digite o codigo do produto: 20
Digite a quantidade em estoque: 35
Informe a quantidade mínima no estoque: 20
*****
```

```
*****
Digite o codigo do produto: 0
Lista:
[{'codigo': 3, 'estoque': 69, 'minimo': 89}, {'codigo': 5, 'estoque': 75, 'minimo': 50}, {'codigo': 2, 'estoque': 43, 'minimo': 45}, {'codigo': 3,
 'estoque': 26, 'minimo': 18}, {'codigo': 20, 'estoque': 35, 'minimo': 20}]
Lista Ordenada:
[{'codigo': 2, 'estoque': 43, 'minimo': 45}, {'codigo': 3, 'estoque': 69, 'minimo': 89}, {'codigo': 3, 'estoque': 26, 'minimo': 18}, {'codigo': 5,
 'estoque': 75, 'minimo': 50}, {'codigo': 20, 'estoque': 35, 'minimo': 20}]

Process finished with exit code 0
```

un | TODO | Problems | Terminal | Python Packages | Python Console | Event Log

33:1 Python 3.8 (AtividadePratica)